

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860 – 1674

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 6(30) CZERWIEC 1988 CENA 150ZŁ

I LOVE ATARI



ATARI ZA DOBRE ŚWIADECTWO

NAD WISŁĄ I WEŁTAWĄ

„Bajtek" ma nowe wcielenie — wydanie „tylko o Commodore". Nieprzypadkowo zbiegło się ono w czasie z polską premierą Commodore, która miała miejsce podczas Międzynarodowych Targów Poznańskich. W chwili, gdy piszę te słowa od czasu przeszłego bardziej logiczny byłby przyszły — wystukuję je bowiem na 5 dni przed otwarciem MTP. Dlatego też o samych Targach napiszemy dopiero najwcześniej za miesiąc. Wtedy też będzie już mniej więcej wiadomo, czy „Tylko o Commodore" zyska wielu nabywców, czy stanie się równie poszukiwane jak nasz „user" poświęcony Atari.

Na marginesie prac nad wszystkimi naszymi zeszytami specjalnymi wciąż staje przed nami pytanie: jak wielką bazę czytelników ma każdy z nich? Nie istnieje dziś, niestety, żadna statystyka ilości poszczególnych typów mikrokomputerów z wyjątkiem sprzedanych za pośrednictwem „Pewexu". Sprzedanie w rekordowo krótkim czasie całego nakładu „Tylko o Atari" sprawiło, że myślimy o większym nakładzie kolejnych numerów. Do optymalnego przyjdzie nam jednak znów dochodzić metodą prób i błędów.

Owa metoda jest pochodną drogi luką trafiają do Polski mikrokomputery. Na ogół jest to tzw. import prywatny, czyli zakup dokonany przy okazji indywidualnego wyjazdu turystycznego lub służbowego. Podczas tego lata z pewnością również trafi do nas kilka, a może i kilkanaście tysięcy nowych komputerów domowych: Zachęcamy ich potencjalnych nabywców do sięgnięcia po portfele przestrzegając jednak: kupujcie sprzęt popularny w naszym kraju, byście mieli dostęp do literatury i oprogramowania. Nie zapominajcie też o tym, że sam komputer, to jeszcze za mało, nawet, gdy chce się tylko pograć w gry! Lepiej zatem, gdy dysponuje się ograniczoną kwotą kupić nawet skromniejszy model komputera, byle np. w zamian nie rezygnować ze stacji dyskietek.

Lato, wakacje, to jednocześnie czas podsumowania kończącego się roku szkolnego. Czytelnicy „Bajtka", mamy nadzieję odebrali mam najlepsze świadectwa. Niestety w tym roku egzaminu z przedmiotu pod nazwą „komputeryzacja" nie zdało całe szkolnictwo. Wciąż jeszcze występują dziwne kłopoty z rozwojem produkcji „Juniorów", z ich systematycznym wykupem przez szkoły. Chociaż my z pewnością nie możemy pomóc szkołom w ich wyposażaniu w sprzęt będziemy starali się wspomóc je na naszych łamach. Pierwszy po wakacjach wrześniowy numer „Bajtka" poświęcić chcemy w głównej mierze zastosowaniu komputerów w edukacji.

Lato, to także okazja do licznych spotkań „Bajtka" z czytelnikami. Jedno z pierwszych w tym roku odbyło się w Dniu Dziecka w... Pradze. Kierownictwo tamtejszego Ośrodka Polskiej Kultury i Informacji postanowiło nas zaprosić dlatego, że właśnie „Bajtek" jest najlepiej sprzedającym się nad Wełtawą polskim pismem. Na spotkaniu było zresztą również tłoczno, a wielu jego uczestników pozostało jeszcze przez kilka godzin, by porozmawiać z nami i skorzystać z fachowych rad obecnych także młodych naukowców z Uniwersytetu łódzkiego, którzy przywieźli do stolicy CSRS swoje komputery i programy. Konkluzja spotkania była prosta: czytelnicy „Bajtka" domagają się jego kolejnych mutacji — tym razem również językowych. Propozycja ta jest z pewnością ciekawa. To przecież będzie okazja nie tylko do zaprezentowania naszego pisma lecz również całego ruchu komputerowego, rodzimych firm i spółek, ludzi zajmujących się w Polsce informatyką. A to się liczy. Eksport słowa może w krótkim czasie zaowocować także eksportem sprzętu, oprogramowania, itp. Zobaczymy, czy najróżniejsze biurokratyczne bariery nie staną nam na drodze. My wierzymy, że któregoś dnia nasz Bajtek stanie się rzeczywiście pismem międzynarodowym. Szkoda byłoby tę szansę zaprzepaścić.

Grzegorz Onichimowski

ZA MIESIĄC:

W kolejnym, 7 numerze Bajtka przeczytacie m.in.:

● wszystko na temat tworzenia, obsługi baz danych na wielkich i małych komputerach, czyli: Vu-File na Spectrum, Synfile na Atari, SEQ, REL i ferajna w klanie Commodore ● tekst drukarki Citizen MSP 15E ● drugie spojrzenie Andrzeja Pilaszka na hanowerski CeBIT ● „Solo Flight", „Stealth" i inne gry ● udamy się także na poszukiwanie zaginionej ciężarówki, przekonamy o tym, czy komputer umie liczyć, założymy wraz z Kubusiem Literką komputerowy notes.

ATARI POJECHAŁO DO KOZIENIC

Rozstrzygnięty został nasz Konkurs Świąteczny ogłoszony w numerze 3/88. Tym razem mimo bardzo wielu listów, jakie otrzymaliśmy prawidłowych odpowiedzi nie było zbyt wiele ok. 2000. Często znajdowaliśmy na Waszych kartkach hasła tak odległe od prawidłowego (Ty też dbaj o dyskietki!), że podejrzewaliśmy ich autorów o zgadywanie z głowy zamiast rozwiązywania. Szczytowym osiągnięciem reprezentantów tej tendencji była propozycja: „Czytaj „Bajtka" od deski do deski". Hasło to, z pewnością słuszne, nie jest niestety prawidłowe, co zdarza się hasłom słusznym niestety dość często.

A oto laureaci konkursu:

Główną nagrodę — komputer Atari 65XE z magnetofonem XC 12 wylosował Paweł Dziewit z Kozienic. Nagrodę tę ufundowało PZ „Karen".

Dwie drugie nagrody, joysticki „Mart" ufundowane przez ich producenta, mgr inż. Tadeusza Trojaka wylosowali Tadeusz Strzoda z Tych i Przemysław Scheffler z Poznania.

Kasety z programami Krajowej Agencji Wydawniczej stały się własnością Piotra Szweda z Krakowa, Jarosława Drewnika z Poznania, Roberta Kitłowskiego z. Warszawy, Macieja Derkowskiego z Kluczborka, Tomasza Łapana ze Świdnika i Stanisława Siuzdaka z Leżajska.

Zwycięzcom gratulujemy, a wszystkich pozostałych naszych Czytelników odsyłamy na przedostatnie strony niniejszego numeru. Rozwiązcie nasz „Konkurs z szesnaściorem", a może teraz szczęście uśmiechnie się do Was.

Bajtek

"BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH" ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21 -12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański- -redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Roman Poznański (z-ca sekretarza redakcji „SM" — sekretarz zespołu „Bajtka"), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

Klany redagują:
Commodore — Klaudiusz Dybowski,
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski,
Atari — Wojciech Zientara, Sergiusz Wolicki.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska

WYDAWCA: RSW „Prasa-Książka--Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 150 zł.
Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA- -KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 010188 n. 150.000 egz.
U-113

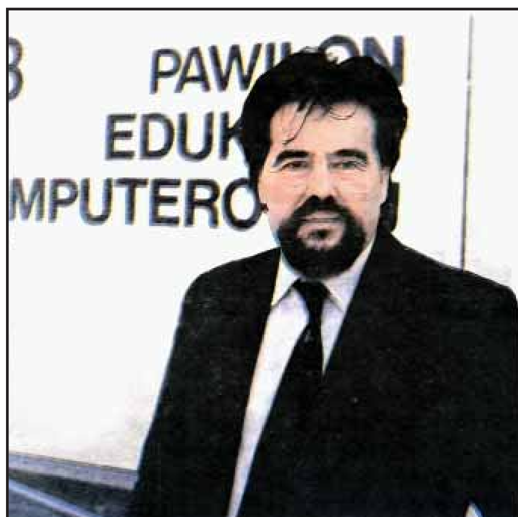




Czy kopiując od kolegi grę miałeś kiedyś wrażenie, że dopuszczasz się nieuczciwości? Jeśli nie, zapraszamy do lektury sondy „Bajtka” poświęconej prawnym aspektom obrotu softwarem w Polsce i ich konsekwencjom.

— SONDA — BAJTKA

NA PIRACKIEJ FALI



ANDRZEJ WIŚNIEWSKI: sekretarz generalny Polskiego Towarzystwa Informatycznego

— Jak Pan ocenia stan prawny, w jakim znalazła się polska informatyka? Czy możemy obecnie chronić swoje programy przed kradzieżą, nie za pomocą sztuczek technicznych lecz również prawnie?

— Teoretycznie możemy. Obowiązujące w Polsce prawo chroni twórczość intelektualną. Nasz kraj, co także chciałem podkreślić, jest sygnatariuszem wielu umów międzynarodowych, które zobowiązują nas do przestrzegania praw autorskich.

— Jeśli tak, czemu nie było u nas spraw sądowych o zagarnięcie czyjegoś programu?

— Nie ma aparatu wykonawczego, który mógłby egzekwować przestrzeganie prawa w tym zakresie. Poza tym w zasadzie ciągle widać, że praca programisty traktowana jest dwojako. Gdy przychodzi np. do płacenia podatków, opracowywanie software'u traktowane jest jako działalność twórcza, Starannie jednak unika się tego sformułowania, gdy ktoś zapyta o prawa autorskie.

— Kto, Pana zdaniem, miałby zatem stać na straży „softwarowego porządku”?

— Stowarzyszenie twórcze takie jak ZAiKS, dysponujące odpowiednią ilością wyspecjalizowanego aparatu, przede wszystkim prawników.

— W pierwszym rzędzie ten komputerowy ZAiKS musiałby zabrać się za proceder kopiowania programów zachodnich. To dziś chyba największy problem - nie tak łatwy do rozwiązania w naszej sytuacji, choćby płatniczej.

— Nie zlikwidujemy żadnymi przepisami kradzieży oprogramowania. Nikt na całym świecie nie doprowadził jeszcze do tego. Trzeba jednak zaprowadzić pewien porządek.

— Jak się do tego zabrać?

— Trzeba po pierwsze ogłosić z określonym dniem „amnestię” dla tych, którzy skopiowali coś już wcześniej. I od tej „godziny zero” ścigać już z całą bezwzględnością zarówno piractwo krajowe jak i międzynarodowe.

— Stać nas na to?

— Spojrzenie: ukradniemy, będzie taniej jest krótkowzroczne. Niestety pojawia się ono oczywiście tylko w rozmowach prywatnych, także z bardzo wysoko postawionymi przedstawicielami naszej administracji. Na długi dystans taka postawa powoduje jednak osłabnięcie aktywności własnych programistów. Możemy sami pozbawić się najlepszych z nich. Wówczas i Zachód zacznie nas przyciskać, każe za wszystko słono sobie płacić, a my nie będziemy mieli żadnych własnych rozwiązań.

— Będzie aż tak źle?

— Sięgnijmy do analogii ze sprzętem. Pojawiają się dziś całkiem poważne głosy: nie produkujmy w ogóle komputerów, przecież możemy je kupić. Dziś nie jesteśmy w stanie dorównać technologiom zachodnim. A zaczęło się niewinnie, od stwierdzenia, że gonić Zachód musimy nie poprzez własne rozwiązania, lecz drogą zakupu licencji bądź zwykłego ściągania efektów pracy innych. Nie chcę podawać konkretnych rozwiązań, przy których sięgnęliśmy po metodę ściągania, ale w dziedzinie sprzętu komputerowego było tego sporo. Zaniedbaliśmy własne ciekawe rozwiązania. I każdy widzi, gdzie się znaleźliśmy. Oby z softwarem nie było tak samo.

ROBERT, „bywalec” giełdy „Bajtka” na ulicy Grzybowskiej

— Jaką widzisz przyszłość przed takimi formami obrotu oprogramowaniem, jakimi stały się giełdy?

— Należy je zlikwidować lub ograniczyć tylko do sprzętu. Tak zwane „wypożyczalnie programów” mogłyby istnieć dalej, pod warunkiem rozprowadzania oryginałów i posiadania na to licencji (podobnie jak video).

— Trochę dziwią takie słowa z ust człowieka, który żyje z handlu na giełdzie. Przecież ciebie to też uderzy.

— Przesada. Ja to przeżyję, gdyż zysk z kopiowania programów, to tylko margines. Chodzę na giełdę, aby nie wypaść z obiegu. Są tam tacy, dla których jest to jedyne źródło dochodów. Dla nich wprowadzenie ochrony prawnej programów będzie poważnym ciosem. Ja obecnie zajmuję się przeróbkami sprzętu (rozszerzenia komputera i stacji) i również piszę programy. To ostatnie jest wyjątkowo nieopłacalnym interesem — jeśli program ma jeden człowiek, to po godzinie ma go cała giełda.

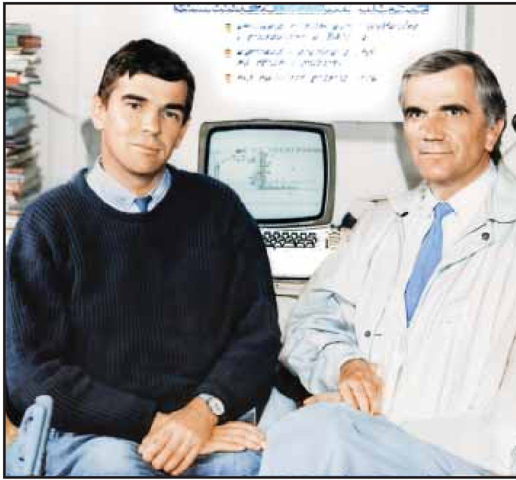
— Czy w takim razie tą większość żyjącą z kopiowania przeniesie się do „podziemia”?

— I będą sprzedawać programy spod płaszcza? To niemożliwe, ten rodzaj handlu wymaga sprzętu, a więc stałego stoiska. Ważny jest inny problem — co będzie piractwem, a co nie. Dam ci przykład. Kupuję legalnie program, pożyczam koledze, który chce sprawdzić, czy warto kupić i kopiuje bez mojej wiedzy. Czy ja również będę odpowiadał? A wykonanie kopii bezpieczeństwa legalnie kupionego programu? Jest tu całe mnóstwo podobnych problemów i trzeba uważać, aby ustawa nie wylała dziecka z kąpielą. Trudno coś powiedzieć dokładniej, nie znamy przepisów obowiązujących na Zachodzie.

— Poza programami na giełdzie można kupić odbitki kserograficzne oryginałów i tłumaczeń różnych instrukcji i podręczników, a także książek wydanych w Polsce.

— To jest uregulowane, są odpowiednie przepisy, również międzynarodowe podpisane przez Polskę. Gdy pierwszy raz przyszedłem na giełdę, byłem zdumiony tym, co się tu wyprawia. Prawie każdy sprzedawca narusza obowiązujące prawo i dawno powinna tu wkroczyć milicja. Na literaturę także trzeba uzyskać licencję lub coś w tym rodzaju Przede wszystkim poziom oferowanych na giełdzie odbitek jest żenujący, a większość z nich ma jeszcze znak lub nazwę firmy. W takiej sytuacji jest to antyreklama. Jest już precedens; sprawa Smurfów i mam nadzieję, że nie będzie to jednostkowy przypadek..

GRA O JUTRO



KRZYSZTOF GAJEWSKI i BOGUSŁAW RADZISZEWSKI — autorzy Warsaw BASICa

W sprawie prawa autorskiego programów możemy wypowiedzieć się z pozycji hobbystów — użytkowników i autorów oryginalnego w skali światowej oprogramowania na najmniejsze ośmiobitowe mikrokomputery firmy Commodore. Hobbyści najczęściej wymieniają się programami. Jednak wymieniać można to, czym się dysponuje i do czego ma się prawa. Sam fakt posiadania nie oznacza dysponowania. Spotkaliśmy w naszej działalności młodych ludzi, którzy chcieli wymienić posiadane przez nich programy (skopiowane od innych) na programy przez nas napisane. Odmowa takiej wymiany kończy się stwierdzeniem: „Ja i tak będę miał te programy tylko trochę później — skopiuję od innych". Takiej wymianie patronują niestety różne instytucje i kluby komputerowe.

Są też studia komputerowe, które pod pretekstem wzięcia w komis egzemplarzy autorskich, udostępniają je do odpłatnego kopiowania. W ten sposób egzemplarze autorskie nie znajdują nabywców. Na giełdach są również sprzedawane nasze programy, oczywiście nielegalnie. Na giełdach warszawskich (m.in. w szkole przy ul. Grzybowskiej) skorzystaliśmy już kilkakrotnie z oferty sprzedaży nam — naszych programów, zabierając dyskietkę bez uiszczenia odpowiedniej opłaty. Programy tam oferowane są „okaleczone" — nie ma na ogół nazwy programu, oraz usunięte są nazwiska autorów. Robią to niestety „miłośnicy" informatyki.

Od ubiegłego roku niektóre instytucje w rozliczeniach z autorami programów stosują reguły zalecane przez Ministerstwo Finansów, a mianowicie: uznają prawa autorskie w zakresie wynagradzania za oryginalne oprogramowanie (uznając to za pracę twórczą). Odbywa się to do tej pory na zasadzie precedensu. Nie ma rozwiązań systemowych w tej dziedzinie. Nie rozwiązuje to również sprawy ochrony praw autorskich. Wydaje się, że jest już pilna potrzeba uregulowania tych kwestii, tym bardziej, że Polska ratyfikowała wiele umów międzynarodowych, które pośrednio dotyczą również wymienionych wyżej spraw. Brak rozwiązań systemowych powoduje, że nawet autorzy Warsaw BASIC nie są traktowani jako twórcy przez radcę prawnego w swojej macierzystej instytucji. Pozostaje to w sprzeczności ze stanowiskiem np. Ośrodka Postępu Technicznego Rady Stołecznej NOT i Izb Skarbowych w Warszawie. OPT uznał naszą pracę za twórczość, w rozumieniu odpowiednich przepisów podatkowych i w związku z tym skorygowano pobrane w ubiegłym roku podatki.

Brak regulacji prawnych w zakresie wynagradzania za działalność twórczą w zakresie programowania i ochrony praw autorskich powoduje swoistą autocenzurę twórców, którzy albo wcale nie rozpowszechniają swoich dzieł, albo puszczają w obieg dzieła okrojone, aby nie wypuścić wszystkich argumentów ze swoich rąk. Gdybyśmy tej „polityki" nie stosowali w swojej działalności, to pierwszy nieuczciwy pośrednik handlowy zawładnąłby całym naszym dorobkiem.



MIROSŁAW MADEJSKI, prezes „Reflexu"

— Co sądzi o ochronie programów szef jednej ze starszych spółek zajmujących się komputerami, prezes „Refleksu"?

— Jest to bardzo złożony problem. Trzy — cztery lata temu zaczęła napływać do Polski nowa generacja komputerów nazwana mikrokomputerami, w tym do zastosowań profesjonalnych Personal Computers. Nowa jakość, nowa technika. Trzeba było nauczyć się uruchamiania , obsługi. Kto myślał o oprogramowaniu? To była fascynacja. Wszędzie powielano instrukcje, jak skopiować program, złamać kod — to była nobilitacja zawodowa programisty. Nawet renomowani eksperci pisali prace na temat małej szansy wygrania procesu o naruszeniu praw przez firmę zagraniczną. Istotnie, nie było nawet procesu.

— Dlaczego zatem obecnie uznano programy komputerowe za efekt twórczy podlegający prawu autorskiemu?

— Dyskusja nad prawem kopiowania programu dotyczy dwu stron zagadnienia. Pierwsza, to moralny i ekonomiczny aspekt kopiowania programów zagranicznych. Z jednej strony dzięki „zdolnościom" polskich programistów mało wydajemy na popularne oprogramowanie związane z mikrokomputerami, lecz z drugiej strony mamy przez to utrudniony lub mało możliwy zakup użytkowych programów profesjonalnych, nie mówiąc już o bardzo specjalistycznych.

Drugi, to prawny aspekt oprogramowania pisanego w Polsce. Dla takich ilości mikrokomputerów, jaką posiadamy, powinna już funkcjonować nowa, właściwa gałąź gospodarki zajmująca się tworzeniem oprogramowania i jego zastosowań. Tak nie jest, moim zdaniem, wyłącznie z powodu braku ochrony praw autorskich.

— Czy to nie jest zbyt daleko idące uogólnienie? Inwestycje na ten cel są też nie mniej ważne.

— Ważne, ale na drugim planie. Każda dziedzina gospodarki jest zdrowa, kiedy po dokonaniu inwestycji na jej powstanie rozwija się na zasadzie- samofinansowania. W informatyce byłoby to możliwe wtedy, gdyby autor lub zespół po opracowaniu pierwszego produktu, sprzedając otrzymałby środki na własne utrzymanie i sfinansowanie kolejnego produktu.

— Przecież jest to kwestia ceny?

— Właśnie. Koszt wytworzenia programu jest bardzo wysoki. Dla zamawiającego jest nazbyt wysoki, aby ta inwestycja mogła spłacić się szybko. Szansa więc jest tylko wtedy, kiedy programy będą pisane tak, aby mogły służyć wielu użytkownikom. Dla sprawiedliwego rozłożenia kosztów twórca sprzedawać je powinien wielokrotnie tym, co osiągną z tego efekt bez obawy, że zastosują go bez uiszczenia zapłaty.

— Jak zatem praktycznie powinna wyglądać ochrona prawna coftwarii?

— Proponuję zapis: „BEZ ZGODY WŁAŚCICIELA ZABRONIONE JEST WYKONYWANIE KOPII PROGRAMU CZY INFORMACJI ZAPISANYCH NA WSZELKICH NOŚNIKACH PAMIĘCI KOMPUTERÓW POD RYGOREM ZWROTU EFEKTÓW I ODPOWIEDZIALNOŚCI KARNEJ"



NORBERT WIESCHALLA, przedstawiciel zachodnioniemieckiej firmy SYNELEC

— Czy w RFN piractwo komputerowe jest problemem podobnym jak w Polsce?

— Z problemem piractwa w dziedzinie software stykamy się obecnie praktycznie we wszystkich krajach, w których istnieje szybko rozwijający się rynek usług komputerowych. Mimo istnienia w RFN ustawodawstwa regulującego odpowiedzialność cywilną i karną wynikającą z pogwałcenia praw autorskich dotyczących oprogramowania, również na rynku zachodnioniemieckim sprawa „taniego software'u:' staje się problemem coraz bardziej palącym.

— Istnienie na rynku oznacza, że nie wszyscy jednoznacznie potępiają nielegalne kopiowanie.

— Jednoznaczna ocena problemu piractwa softwarowego jest chyba niemożliwa. Z jednej strony bowiem istnienie taniego, łatwo dostępnego software^ umożliwia korzystanie z niego wielu małym firmom, w tym także prywatnym posiadaczom komputerów. Z drugiej strony, żadna z poważnych firm komputerowych, jak również użytkowników oprogramowania w RFN nie zakupi oprogramowania pirackiego, niekompletnego, z którym mogłaby mieć jakiekolwiek kłopoty. Oprogramowanie, jak również hardware, musi się w naszych warunkach spłacać w jak najkrótszym czasie, a jest to możliwe tylko wtedy, gdy jest ono w pełni sprawne, a pracownicy korzystający z niego odpowiednio przeszkoleni.

— Jakie dla Polski wnioski wynikają z doświadczeń RFN.

— Moim zdaniem istnieje pilna potrzeba uregulowania prawnego, żywiotowo i w niekontrolowany sposób rozwijającego się w Polsce rynku softwarowego. Określenie pewnych ram praw-

nych powinno rynek ten uporządkować. Jestem realistą i wiem, że wprowadzenie takiego ustawodawstwa nie zlikwiduje problemu piractwa softwarowego, uprawianego zwłaszcza przez tak dobrych informatyków, jak polscy.



KRZYSZTOF KACPRZAK,
wiceprezes spółki Inter-Ams

— Czy wprowadzenie podobnych do zachodnich, reguł ochrony praw autorów programów komputerowych może się Polsce per saldo opłacać — firmom, programistom, krajowi?

— Zawsze uważałem tworzenie softwaru za ;wego rodzaju produkcję. Firmy komputerowe powinny żyć przede wszystkim z produkcji oprogramowania, przynajmniej firmy polskie. Nie jesteśmy cowiem w stanie „konkurować"'z Zachodem w dziedzinie produkcji sprzętu.

— Twoja teoria nie pasuje do praktyki. Zarabiacie chyba głównie na sprzęcie.

— Akurat my nie, my żyjemy z produkcji programów. W przyszłości chcemy handlować tylko oprogramowaniem. Oczywiście brak jasnych reguł dotyczących prawa do programów utrudnia nam życie. Pomagamy sobie nie poprzez skomplikowane sposoby zabezpieczeń, lecz poprzez dość skrupu- atną ewidencję sprzedawanego softwaru. Gdy program nasz zacznie się rozchodzić poza naszą kontrolą będziemy w stanie chociaż stwerdzić, kto go od nas kupił jako pierwszy.

— Czy to może być skuteczne?

— Zabezpieczanie takich programów, jak system technicznego przygotowania produkcji, jest bezsensowne. W obecnej sytuacji prawnej musimy jednak sięgać po jakieś narzędzia. To, które wymieniłem jest z pewnością niedoskonałe, ale jednak coś robić trzeba.

— Co musiałoby się zmienić by software stał się u nas rzeczywiście zyskowanym biznesem.

— Nie tyle chyba prawo, co relacje cenowe. Człowiek, jego myśl twórcza ukryta w programie, wciąż są jeszcze u nas mniej warte od sterty elektronicznego „złomu". Natomiast nie wierzę w skuteczność walki z komputerowym piractwem.

OD REDAKCJI

Podsumowanie ankiety nie miałoby większego sensu — wszyscy uczestniczący w niej mieli taką samą opinię. Twórczość programistów należy chronić tak samo jak chroni się efekty pracy pisarzy czy grafików. Ciekawe dlaczego zabroniono wypożyczania nielegalnych kopii filmów, nie stworzono zaś żadnych przepisów uniemożliwiających taki sam proceder dotyczący programów? Może dlatego, że w tym przypadku w grę wchodzą nie tylko zastosowania domowe, iecz przede wszystkim software użytkowy miliardowej wartości. Głosy z naszej ankiety przekonują, że takie „ekonomiczne" myślenie oparte jest na nader kruchych podstawach.

— MINI — PACMAN

Część Czytelników interesują wyłącznie gry, inni chcieliby więcej informacji o programowaniu, a „Bajtek" nie jest z gumy. Aby wilk był syty i owca cała, drukujemy grę napisaną w Action!.

Po ukazaniu się „Bajtka-Atari" do redakcji nadeszło wiele listów z pytaniami o język **Action!.** Równolegle do prowadzonego w wydaniach specjalnych „Bajtka" kursu tego języka będą więc w klanie Atari prezentowane napisane w nim programy. Pierwszym jest prosta gra zręcznościowa oparta na motywach popularnego „PacMana".

Zadaniem gracza jest zbieranie kropek na planszy. Należy przy tym unikać wygłodniałego potwora, który chciałby zjeść PacMana na obiad. Po zebraniu wszystkich kropek następuje przejście na wyższy poziom (razem jest ich 20). Oczywiście do poruszania się po planszy trzeba używać joysticka.

Jeszcze krótka informacja dla osób, które nie znają **Action!**. Po wpisaniu programu trzeba przejść do monitora przez wciśnięcie klawiszy CTRL-SHIFT-M. Teraz poleceniem C kompilujemy program i zapisujemy wersję skompilowaną przez S C: (kaseta) lub S D:PACMAN.OBJ (dyskietka). Program źródłowy można również zapisać (przed wejściem do monitora) przez wciśnięcie klawiszy CTRL- SHIFT-W i podanie nazwy C: lub D:PACMAN.ACT.

Po skompilowaniu lub wczytaniu skompilowanego programu uruchamiamy go poleceniem R (z monitora). Po zakończeniu gry następuje powrót do monitora i wyświetlenie meldunku błędu „Error:128". Jest to objaw prawidłowy i nie należy szukać btędu w programie. Ten błąd jest powodowany przez procedurę Break(), przerywającą program.

Instrukcje w **Action!** mogą byc oddzielone od siebie spacją lub dwukropkiem. W jednej linii można umieścić dowolną liczbę instrukcji. Rozmieszczenie instrukcji w programie jest więc dowolne, a wcięcia w wydruku służą jedynie poprawieniu jego czytelności.

Program nie jest skomplikowany i nawet początkujący programista łatwo go zrozumie po dokonaniu analizy. Trzeba przy tym pamiętać, że program w **Action!** jest ZAWSZE wykonywany od ostatniej procedury.

Wojciech Zientara

```
;Min-PacMan
;(c) 1988,  Bajtek
;written in Action!

MODULE
BYTE level,crsinh=752
CARD score

PROC Pause(BYTE  j)
CARD k

 FOR  k=0 TO j*j DO OD
RETURN

PROC Playfield()
BYTE colpf=710,cclpm=709, i
 Graphics(0)
 colpf=192, cclpm=206, crsinh=1
 Position(14,2)
 Print("Mini PacMan")
 Position(9,5)
 Print ("ðÝÝÝÝÝÝÝÝÝÝÝÝÝ¸")
FOR i=6 TO 12
 DO
  Position(9,i)
  Print("|.............|")
 OD
Position(9, 13)
 Print ("³ÝÝÝÝÝÝÝÝÝÝÝÝÝ˙")
Position(15,14)
PrintE("WYNIK:   ")
Position(15, 4)
Print("POZIOM: ")
PrintBE(level)
RETURN
BYTE FUNC GetLevel()

BYTE i

DO
 Open (1,"K:",4,0)
 i=GetD(1)
 Close(1)
UNTIL  (i>48 AND  i<58)
OD
RETURN(i-48)

PROC Setup()

 Graphics(0) crsinh=0
 Position(5,5)
 Print("POZIOM  TRUDNOSCI (1-9) ?")
 level=GetLevel()
 crsinh=1 PutE()
 score=0
RETURN

PROC End(CARD score)

BYTE i
FOR i=0 TO 15
DO
 Sound(1,10*(15-i),2,(15-i))
 Pause(40-i)
OD
SndRst()
Graphics(0) crsinh=1
Position(15,8)
Print("WYNIK:   ")
PrintCE(score).
Pause(200)  crsinh=0
Position(11,14)
Print("Jeszcze raz (T/N)?")
Open(1,"K:",4,0)
i=GetD(1)
Close(1)
IF i=78 OR  i=110 THEN
 Break()
FI
RETURN

PROC Game(BYTE  level)
BYTE xp,yp,dx,dy,xm,ym,a,b,i,s,c,q
FOR q=level  TO 20 DO
Playfield()
xp=10 yp=6 xm=29 ym=12
i=0
DO
a=Locate(xp,yp)   b=Locate(xm,ym)
Position(xp,yp)   Put(188)
Position (xm,ym)  Put(162)
Pause((20-level)*5)
s=Stick(0)
IF a=46 THEN
 Sound(0,100,10,10)
  a=32 i==+1 score==+1 FI
IF s=7 AND xp<29 THEN dx=2
ELSEIF s=11 AND xp>10 THEN dx=0
ELSE dx=1
FI
IF s=13 AND yp<12 THEN dy=2
ELSEIF s=14 AND yp>6 THEN dy=0
ELSE dy=1
FI
xp==+dx-1  yp==+dy-1
Position(22,14)   PrintC(score)
SndRst()
Position(xp-dx+1,yp-dy+1) Put(a)
Position(xm,ym) Put(b)
c=Rand(10)
IF c<level THEN
IF xp>xm THEN xm==+1
ELSEIF  xp<xm THEN  xm==-1
FI
IF yp>ym THEN ym==+1
ELSEIF yp<ym THEN ym==-1
FI
FI
IF xp=xm AND yp=ym THEN RETURN
FI
UNTIL i=140 OD
level==+1
OD
RETURN

PROC Pacman()

DO
 Setup()
 Game(level)
 End(score)
OD
RETURN
```

SPOSOBY OCHRONY

Temat zwalczania piractwa komputerowego jest tak stary jak same komputery. W krajach o rozwiniętej komputeryzacji istnieją odpowiednie przepisy służące ochronie praw autorskich programistów. Nawet najdoskonalszy przepis nie zapobiegnie jednak „koleżeńskiej" wymianie programów. Dlatego też wymyślono różne zabezpieczenia mające uniemożliwić kopiowanie.

Programy dla komputerów Atari występują w trzech formach: na kasecie, na dyskietce lub w module ROM (cartridge). Każda z tych form ma specyficzne cechy i wymaga odrębnego sposobu zabezpieczania. Opiszę je więc kolejno, a na końcu przedstawię sposoby ochrony wspólne dla wszystkich nośników.

PROGRAMY KASETOWE

Początkowo wystarczyło, że programy kasetowe (i dyskowe) uruchamiały się samoczynnie po wczytaniu oraz były odporne na klawisze BREAK i RESET. Jest to bardzo proste do wykonania, a nie pozwala na skopiowanie. Z chwilą pojawienia się pierwszego programu kopiującego sprawa znacznie się skomplikowała.

Bez zagłębiania się w szczegóły wystarczy powiedzieć, że zabezpieczenie programów kasetowych polega na zapisie w blokach o niestandardowym formacie. Pierwszy blok zawiera wtedy program modyfikujący procedury odczytu z magnetofonu i odczytujący pozostałe bloki. Modyfikacje te są bardzo różne i doskonalenie programów kopiujących nie zawsze nadąża za ich rozwojem.

Niestety, program nagrany na kasecie zawsze można skopiować przy użyciu dwóch zwykłych magnetofonów. Między innymi dlatego na kasetach mniej jest programów i zwykle wersje kasetowe są uboższe od dyskowych.

CARTRIDGE

Moduł ROM (cartridge) jest to układ pamięci EPROM (czasem ROM), w którym zapisany jest program. Najprostszą metodą skopiowania takiego programu jest przepisanie go do zbioru, który wczytuje się w to samo miejsce w pamięci, jakie zajmuje cartridge. Ochrona przed tym sposobem jest stosunkowo prosta. Wystarczy, że program wpisuje coś w obszar, który sam zajmuje. Do pamięci ROM nie da się nic wpisać, więc oryginał będzie działał normalnie. Natomiast ten sam program wczytany z kasety lub dyskietki do pamięci RAM komputera będzie uszkadzał sam siebie.

Program zapisany na cartridge'u można także skopiować przepisując jego zawartość do innego EPROM-u. Wymaga to jednak posiadania programatora i ze względu na koszt układów EPROM jest stosunkowo drogie. Poza tym skopiowany program jest nadal zabezpieczony przed dalszym kopiowaniem i to w tym samym stopniu, co oryginał.

Dodatkową zaletą jest trudność uszkodzenia takiego programu oraz skrócenie do minimum czasu potrzebnego na jego uruchomienie. Z tych powodów rozpowszechnianie programów na cartridge'ach jest na razie najlepszym sposobem ich zabezpieczania. Może właśnie dlatego tak mało jest w Polsce programów w tej formie.

PROGRAMY DYSKOWE

Pierwsze zabezpieczenia programów zapisanych na dyskietkach polegały na uniemożliwieniu skopiowania przy pomocy DOS-u. DOS 2.0, w którym zapisana była większość programów, posiada funkcje kopiowania zbiorów (C — Copy file i O — Duplicate file) oraz funkcję powielania dyskietki (J — Duplicate disk). Wszystkie te funkcje korzystają ze spisu zawartości (directory) i mapy sektorów (VTOC).

Sposób ochrony programu narzuca się sam. Wystarczy przenieść directory i VTOC w inne miejsce i standardowy DOS już nic nie odczyta. Wykonanie tej operacji polega na zmianie zawartości czterech bajtów DOS-u i jest prymitywnie proste. Zabezpieczony w ten sposób program może składać się z wielu części, a nawet może zapisywać coś na własnej dyskietce oraz na dyskietkach sformatowanych przez siebie.

Gdy powstały programy kopiujące dyskietkę sektor po sektorze, zabezpieczenie to przestało być skuteczne. Wymyślono więc wadliwe sektory (znane powszechnie pod angielską nazwą „bad sectors"). Sektor taki zapisywany przy zmniejszonej prędkości obrotowej napędu dyskowego, co powoduje gęstsze zapisanie poszczególnych bajtów. Na normalnej stacji nie da się tego wykonać i oczywiście odczytać. Chroniony program po uruchomieniu próbuje dokonać odczytu tego sektora. Jeżeli się nie uda, to wszystko jest w porządku. Jeśli tak, to znaczy, że dyskietka nie jest oryginalna i program zawiesza się.

Oczywiście i na to jest rada. W odpowiednich miejscach na dyskietce należy zapisać wadliwe sektory i kopia działa. Dokonać tego można po drobnej przeróbce stacji dysków — może to zrobić każdy, kto umie się posługiwać lutownicą i wie jaka to przeróbka. Pojedyncze wadliwe sektory można zapisać nawet bez takiej przeróbki.

Ostatnim, najnowszym pomysłem są tzw. sektory nieokreślone (misassigned sectors). Aby zrozumieć na czym to polega, trzeba nieco bliżej przyjrzeć się dyskietce. Powierzchnia dyskietki jest podzielona na współśrodkowe okręgi zwane ścieżkami. Z kolei każda ścieżka podzielona jest na odcinki, które nazywamy sektorami. Na dyskietce zaformatowanej przy pomocy DOS 2.0 jest ich 18. Poza tym na ścieżce znajduje się jeszcze jeden mniejszy odcinek, w którym zapisany jest numer ścieżki, znak jej początku i inne ważne dla stacji informacje.

Każdy sektor zawiera 128 bajtów danych (programu), które można normalnie zapisywać i odczytywać oraz 44 bajty identyfikacyjne, które są wykorzystywane wyłącznie przez wewnętrzny kontroler stacji dysków. Te 44 bajty stanowią klucz do nieokreślonych sektorów. Zawierają one następujące informacje:

1. numer sektora
2. numer ścieżki
3. sumę kontrolną sektora (CRC)
4. znak początku danych (data mark)
5. znak zapełnienia sektora

Sektory nieokreślone są tworzone przez zmianę bajtów identyfikacyjnych w sposób niemożliwy do uzyskania w normalnej stacji dysków. Ze względu na dokonaną zmianę rozróżniamy trzy rodzaje nieokreślonych sektorów: z błędną sumą kontrolną (CRC error), z błędnym znakiem danych (bad data mark) i sektory podwójne (duplicate sectors).

Zastosowanie dwóch pierwszych rodzajów nieokreślonych sektorów jest podobne do sektorów wadliwych, lecz zawartość sektora jest przy tym odczytywana Program może więc sprawdzać błąd przy odczycie sektora, jak i poprawność jego zawartości. Nieco inaczej działają sektory podwójne.

Normalne sektory na ścieżce mają kolejne numery od 1 do 18. Z sektorami podwójnymi mamy do czynienia wtedy, gdy numery sektorów zostaną zmienione w ten sposób, że dwa z nich mają taki sam numer. Załóżmy, że na dyskietce są dwa sektory o numerze 10. W jednym z nich umieszczona jest jakaś informacja (np. część programu), a w drugim same zera (lub inne informacje). Zabezpieczony w ten sposób program odczytuje sektor numer 10 i umieszcza go w buforze, następnie odczytuje po raz drugi sektor 10 i zapisuje w innym buforze. Teraz bufory te są porównywane. Jeśli zawartość ich jest różna, to program jest kontynuowany. Gdy oba bufory mają jednakową zawartość, to znaczy, że dyskietka nie jest oryginalna i program się zawiesza.

Skopiowanie tak zabezpieczonych programów jest również możliwe. Konieczna jest jednak do tego celu specjalnie przerobiona stacja dysków. Musi ona posiadać zmieniony system operacyjny, a to wymaga wymiany układów pamięci stałej (ROM) wewnątrz stacji.

ZABEZPIECZENIA PROGRAMOWE

Zupełnie odrębną metodą, niezależną od zastosowanego nośnika, jest zabezpieczenie programu tzw. kluczem. Jest to kilka zwykłych rezystorów zamontowanych w standardowej wtyczce do joysticka i zalanych żywicą. Przed uruchomieniem programu należy taki klucz umieścić w gnieździe joysticka. Program odczytując wewnętrzne rejestry joysticków i potencjometrów sprawdza, czy w gnieździe znajduje się właściwy klucz.

Sposób ten jest dość niewygodny, gdyż producent musi do każdego klucza dopasować wartości w programie. Zwiększa to także koszty, a więc i cenę programu. Na dodatek stosunkowo łatwo można odczytać wartości podawane przez klucz przy pomocy prostego programu w Basicu i wykonać jego kopię. Oczywiście można wyprodukować klucz, który działałby dopiero z właściwym programem, lecz koszt jego byłby bardzo duży.

Innym sposobem ochrony programowej jest wpisywanie w jakimś miejscu programu (lub w kilku miejscach) jego numeru seryjnego. Nie zabezpiecza to wprawdzie przed skopiowaniem, lecz pozwala na natychmiastową identyfikację pirata. Sposób ten jest jednak kłopotliwy, gdyż wymaga prowadzenia dokładnej rejestracji wszystkich nabywców. Z tego właśnie powodu jest bardzo rzadko stosowany. W Polsce, ze względu na brak ochrony prawnej oprogramowania, wykrycie osobnika, który skopiował program, ma tylko znaczenie moralne.

Wszystkie wcześniej opisane zabezpieczenia mają jedną zasadniczą wadę. Przy pewnej znajomości języka, w jakim został napisany program, można dokonać zmian w tym programie i umożliwić jego działanie nawet po skopiowaniu. Aby to utrudnić często stosuje się ukrywanie treści programu. Na przykład najpierw wczytywany jest co drugi bajt, a reszta uzupełniana jest później albo właściwy kod uzyskuje się po wykonaniu operacji EOR, AND i/lub OR na odczytanych danych. Oczywiście przedłuża to tylko czas potrzebny na skopiowanie, lecz nie uniemożliwia go.

PERSPEKTYWY

Podstawowe pytanie brzmi następująco: czy można stworzyć program całkowicie niekopiowalny? Oczywiście odpowiedź musi być negatywna — takiego programu nie ma i nie będzie. Na każdy rodzaj zabezpieczenia znajdzie się jakiś sposób, jest to tylko kwestią czasu. Z drugiej strony zabezpieczenia uniemożliwiają legalnemu nabywcy wykonanie kopii bezpieczeństwa. W takim razie po przypadkowym zniszczeniu lub uszkodzeniu programu trzeba się zwracać do producenta o wymianę. Poza tym niektóre zabezpieczenia utrudniają korzystanie z programu. Użytkownicy stacji dysków LDW Super 2000 wiedzą, że nie wszystkie programy można z tej stacji uruchomić. Posiada ona własną obsługę błędów i często zawiesza się po napotkaniu na zabezpieczenia. Trzeba również pamiętać, że zabezpieczenie programu także kosztuje i w pewnej chwili może dojść do sytuacji, w której zabezpieczenie będzie droższe niż zabezpieczony program. Dlatego wielu programistów rezygnuje całkowicie z zabezpieczeń. Do tego potrzebne jest jednak przede wszystkim odpowiednie prawo chroniące ich interesy.

Wojciech Zientara

TAJEMNICE ATARI (7)

Na wstępie chcielibyśmy podziękować Czytelnikom za listy pomagające redagować nam „Tajemnice Atari". Mamy jednak prośbę. W listach cytujecie często fragmenty instrukcji, a nasza rubryka ma trochę inne zadanie. Chcielibyśmy, aby zamieszczane tu sztuczki czy sposoby ułatwiały ukończenie lub kontynuację gier. Dlatego nie interesują nas klawisze funkcyjne czy opisy tego, co dzieje się na ekranie. Od tego są instrukcje do gier. W „Tajemnicach Atari" szukamy raczej „lekarstwa na zwycięstwo", czyli łatwiejszej, szybszej drogi do sukcesu.

Oczywiście nie uzurpujemy sobie prawa do nieomylności, dlatego też dziękujemy Mirkowi Kanii za poprawki do zamieszczonej w czwartym odcinku „Tajemnic" recepty na ukończenie gry **NINJA** (Mastertronic). Aby zostać tytułowym bohaterem należy zdobyć sześć bożków (Idols), następnie dotrzeć do najwyższej komnaty. Po zwycięskiej walce z pięcioma karatekami i złym Ninją wejść w posiadanie ostatniego siódmego bożka. Po powrocie z tymi zdobyczami do Torii in the See (początek gry) zostaniesz zwycięzcą gry. Dziękujemy! **BOUNTY BOB STRIKES BACK** (Big Five Software) jest jedną z najlepiej „dopracowanych" gier, ciekawa grafika, wiele plansz i atrakcyjna fabuła pozwalają zaliczyć ją do komputerowej elity (choć przykre, że nie uruchamia się ona na ATARI 130 XE). Mnogość pułapek i wrogich, maszerujących bomb stawia przed graczem bardzo trudne zadanie do wykonania. Autorzy gry chcieli pomóc tym mniej cierpliwym i niezbyt wytrwałym. Po wczytaniu programu wystarczy nacisnąć klawisz OPTION, aby mieć możliwość ustawienia joystickiem dowolnych opcji gry. Ostatnia z nich to pole do wpisywania specjalnych kodów (Special Code). W polu tym należy wprowadzić wartość 61800. Ruch joystick'a w lewo zmniejsza wartość o 1, w prawo zwiększa o tę samą wielkość, jeżeli ten ruch połączymy z naciśnięciem FIRE to zmieniać będziemy nie jednostki a tysiące. Teraz wystarczy nacisnąć tylko następujące klawisze, najpierw [F], potem OPTION i START. Rozpocznie się pozornie normalna gra, ale tylko tak nam się wydaje, gdyż naciśnięcie:

[F] — spowoduje że Bob zostanie zaopatrzony w tajemną moc latania, a poruszające się miny staną się nieszkodliwe, ponowne naciśnięcie klawisza [F] pozwoli powrócić do normalnej gry. Jeżeli teraz nasz biedny Bob będzie spadał i niechybnie grozić mu będzie rozbicie, wystarczy ponownie nacisnąć [F] i spokojnie sprowadzić go na powierzchnię gdzie znajdzie oparcie pod stopami.

[Q] — ten klawisz należy nacisnąć gdy znudzi się nam zabawa na tym poziomie, Bob bez względu na to w jakiej znajduje się sytuacji rozpocznie zabawę od następnego ekranu.

[A] — powoduje powrót do poprzedniego poziomu.

[B] — Bob powraca do swojej normalnej pozycji.

[Z] — resetuje ekran i pozwala na zabawę od nowa na tym samym poziomie.

MINER 2049'er (Big Five Software) to „młodszy brat" opisywanego wcześniej Bounty Boba. Ten sam bohater, przygody co prawda trochę inne, ale wcale nie mniej niebezpieczne. Poważne kłopoty z reguły zaczynają się na trzecim etapie, gdzie mimo wielu prób nie udaje się nam dostać na półkę gdzie znajduje się żelazko. A sprawa jest banalnie prosta. Prostokąty z cyframi 1, 2 i 3 to po prostu windy. Jak wejść do windy wiadomo, ale co dalej? Zastanówcie się przez chwilę... co robi każdy człowiek gdy znajdzie się w windzie... tak jest! Naciska guzik z wybranym numerem piętra, więc zróbmy to samo, naciśnijmy [3] i pojedziemy na trzecie piętro, tam gdzie tak usilnie próbowaliśmy się dostać. Jest jeszcze sposób, aby w ogóle nie brać udziału w tym etapie gry. (Uwaga poniższa dotyczy wersji dyskowej). Zanim wgramy „MINERA" musimy załadować Translator (np. FIX XL), a dopiero potem gry. Na ekranie ukażą się wtedy opcje do wyboru, używając joystick'a będziemy mogli ustawić numer stacji (stations) i strefy (zone) od których chcemy rozpocząć grę oraz liczbę możliwych niepowodzeń (lives). Należy to robić bardzo uważnie i ostrożnie gdyż ustawiane wartości nie będą widoczne na ekranie (każdy ruch joystick'a w górę zwiększa wybraną opcję o 1).

BEACH HEAD (Access Software) jest grą, która kusi fabułą, możesz przecież na pewien czas wejść w rolę komandosa w akcji. Uciekasz przed minami w zatoce. Walczysz ze statkami i samolotami na morzu, prowadzisz walkę na lądzie kierując czołgiem. Niestety rzeczywistość, nawet ta komputerowa często bywa zbyt trudna, a dotarcie na ląd okazuje się niewykonalne. Nie ma się o co martwić. Jest na to prosta rada. Po wczytaniu gry nie rwijmy się od razu do joystick'a lecz włóżmy go spokojnie do pierwszego gniazda i czekajmy. Po chwili rozpocznie się demonstracja możliwości programu, będziemy mogli obejrzeć poszczególne fazy walki, teraz należy tylko wybrać moment, który nam najbardziej odpowiada itrzymając joystick w ręku włączyć się do gry.

CRYSTAL CASTLES (Atari Software) to jeden z programów z trójwymiarową grafiką. Sympatyczny ludzik ma poważne kłopoty, aby pokonać poszczególne komnaty kryształowych zamków. Na pierwszą z nich jest pewien niezbyt skomplikowany sposób, wystarczy, że skierujemy go w dolny lewy róg i 'naciśniemy FIRE. Nasze konto wzbogaci się o 100 000 punktów a bohater gry znajdzie się w następnej komnacie, a dalej, dalej musicie radzić sobie sami.

Tomasz Mazur
Sergiusz Piotrowski

WATSON

Prawie każdy posiadacz stacji dysków choć raz próbował zbadać, co też właściwie jest na dyskietce. Do tego celu służy wiele różnych programów, m.in. „Disk Wizard", „Disk Scanner" i „Sherlock". Każdy z nich ma swoje wady i zalety. Także w Polsce powstał podobny program, który na przekór „Sherlockowi" otrzymał nazwę „Watson".

Już na pierwszy rzut oka „Watson" wyróżnia się spośród innych programów dzięki oparciu całego programu na szeroko rozbudowanej technice okien. Wszystkie funkcje mają własne okna ekranowe, przy czym niektóre z nich są rozwijane w razie potrzeby. Po zakończeniu pracy w tych oknach ich zwinięciu przywracana jest poprzednia zawartość ekranu.

Program umożliwia przede wszystkim wczytanie dowolnego sektora i redagowanie jego zawartości (w postaci liczb szesnastkowych, znaków kodu ASCII lub znaków kodu wewnętrznego). Poza redagowaniem w dowolnie wybranym systemie, drugą, rzadko spotykaną możliwością jest przesuwanie zawartości redagowanego sektora klawiszami CTRL-INSERT i CTRL-DELETE tak, jak w normalnym edytorze ekranowym. Oprócz tego program wyposażony jest w miniasembler podający menemonik rozkazu, na którego kodzie znajduje się kursor.

Do współpracy ze stacją program korzysta z własnych procedur I/O, dzięki czemu odczyt sektorów wstecz (od wyższych do niższych numerów) jest szybszy niż w normalnej, rosnącej kolejności. Zależnie od trybu pracy sektory wczytywane są w kolejności numerów lub według następstwa w zbiorze.

Dalsze możliwości dostępne są po wywołaniu pomocniczego menu. Zawiera ono następujące funkcje: odczyt directory, zmiana trybu pracy (dyskowy lub plikowy) przeszukiwanie dysku, zmiana trybu wyświetlania (ATASCII lub internal), zamiana liczb HEX-DEC, mapa dyskietki, przegląd zestawu znaków, zapis sektora i oczywiście wyjście do DOS-u.

Nie ma tu miejsca na opis programu więc ograniczę się tylko do kilku



ciekawych szczegółów. Przy zapisie sektora najpierw jego zawartość jest wczytywana do oddzielnego bufora, co stanowi dobre zabezpieczenie przed błędami. Zawartość tego bufora może być oczywiście przeniesiona do bufora głównego.

Bardzo dobrze została rozwiązana zamiana liczb szesnastkowych na dziesiętne i odwrotnie. Natychmiast po wpisaniu dowolnej cyfry w jednym z tych systemów pojawia się równowartość w drugim systemie. W ten sam sposób rozwiązane jest wprowadzanie wzoru do przeszukiwania dyskietki (jednocześnie w kodzie szesnastkowym wewnętrznym i ATASCII).

Obsługa programu jest bardzo prosta i łatwa do zapamiętania. Do programu dodawana jest również skrócona instrukcja w postaci pliku tekstowego. Na zakończenie jeszcze jedna ciekawostka. Po przejściu do DOS-u pojawia się ekran z dokładnie taką samą zawartością, jak w chwili wywołania „Watsona".

Autor: JBW Softwork i CHAOS Software
Dystrybutor: Klub Studentów Politechniki Rzeszowskiej „PLUS"

Wojciech Zientara



ATARI XE GAME

Ponad 20 milionów sprzedanych egzemplarzy gier komputerowych skłoniło konstruktorów z firmy Atari do opracowania wyspecjalizowanego urządzenia, którego głównym zadaniem jest właśnie dostarczanie rozrywki.

Atari XE-Video-Game-System zawiera wszystkie elementy niezbędne dla każdego gracza, niezależnie od wieku. System przystosowany jest do zmagań w grach sportowych, strategicznych i zręcznościowych, które narzucają sprzętowi konieczność spełnienia różnych wymogów.

Najważniejszą i najciekawszą część składową stanowi 64 KB komputer, spełniający nie tylko rolę centrali sterowniczej w grach przygodowych. Może on z powodzeniem pomóc w zarządzaniu danymi w domowym archiwum płyt czy kaset lub zastąpić maszynę do pisania, jeśli zechcemy przygotować zaproszenia na prywatkę. Zachęca on do twórczej rozrywki, ponieważ wbudowany Atari Basic jest niezłym narzędziem do samodzielnego zaprogramowania gry lub innego programu. Gwarantuje on uzyskanie dobrej grafiki na ekranie zwykłego telewizora lub monitora. Nie zapomniano również o bogatej gamie urządzeń zewnętrznych skonstruowanych z myślą o komputerach Atari. Współpraca z drukarką, magnetofonem lub stacją dysków jest możliwa dzięki istniejącym złączom szeregowym.

Oddzielona od klawiatury konsola sterownicza dla gracza zawiera gniazdo do dołączenia zewnętrznego modułu ROM z grami (cartridge'a). Do wyposażenia standardowego należy joystick. Skonstruowany został ergonomicznie, z myślą od długotrwałym użytkowaniu nawet przez gracza o ogromnym temperamencie. Ciekawą zabawką znajdującą się w zestawie jest pistolet przydatny w grach zręcznościowych.

Atari XE Video-Game-System wszechstronnie spełnia wymagania stawiane komputerowej zabawce. Przeznaczono go głównie dla rozrywki jednego lub kilku grających, ale zainstalowanie języka programowania świadczy o poważnym traktowaniu również i ambitniejszych użytkowników. Urozmaicony wybór gier pozwala każdemu wybrać coś odpowiedniego dla siebie.

DANE TECHNICZNE:

Pamięć
- 64 KB RAM
- 24 KB ROM
- Możliwość dołączenia zewnętrznego modułu ROM (cartridge'a).

Grafika
- 320 x 192 punkty
- 11 trybów graficznych oraz 5 tekstowych z możliwością wzajemnej kombinacji.
- 256 kolorów
- Płynne przesuwanie ekranu w pionie i poziomie
- Grafika graczy i pocisków z rozpoznawaniem kolizji obiektów
- Przerwania wywoływane przez program ANTIC-a

Budowa wewnętrzna systemu
- Jednostka centralna: 8-bitowy mikroprocesor 6502C, częstotliwość zegara 1,79 MHz (USA) lub 2.21 MHz (Europa).
- Specjalizowane mikroprocesory do przetwarzania obrazu i dźwięku.
- Selftest umożliwiający kontrolę systemu i wykrycie uszkodzenia.

Złącza
- Szeregowe do połączenia urządzeń peryferyjnych
- Gniazda przyłączeniowe dla joysticka, pistoletu laserowego
- Gniazdo zewnętrznego modułu ROM (cartndge'a)

Oprogramowania
- Wbudowany Atari Basic ze specjalnymi instrukcjami graficznymi i dźwiękowymi
- Różne języki programowania: Logo, Pascal, Pilot, Assembler i inne
- kompatybilność oprogramowania z Atari 800XL/65 XE
- zainstalowana gra

Dźwięk
- Cztery niezależne, programowalne generatory dźwięku
- Zakres częstotliwości od 0,6 Hz do 800 kHz

(j.j.)

ZOSTAŃ NIEŚMIERTELNYM

Przed podaniem następnych poprawek umożliwiających uzyskanie nieśmiertelności lub innych „korzyści" w grach przypominamy, że sposób ich wprowadzenia był opisany w poprzednim odcinku ("Bajtek" 5/88).

W popularnej grze **Crystal Castle** komórki $1BB8 i $1BBB zawierają liczbę żyć każdego z graczy. Dla uzyskania nieśmiertelności trzeba więc odnaleźć rozkaz DEC $1BB8,X ($DE, $B8, $1B; dziesiętnie 222, 184, 27; w kodzie ASCII: znak „^" w negatywie, „8" w negatywie i znak ESC) i zamienić go na LDA $1BB8,X. Oznacza to zamianę pierwszego bajtu tej sekwencji na $BD (dziesiętnie 189, w ASCII: „ =" w negatywie).

Liczba żyć w grze **Robin Hood** jest zapisana pod adresem $2DA5. Należy więc odnaleźć rozkaz DEC $2DA5 ($CE, $A5, $2D; dziesiętnie 206, 165, 45; ASCII: „N" w negatywie, „%" w negatywie i „-") i zamienić pierwszy bajt na $AD (dziesiętnie 173, ASCII: „-" w negatywie), co da w wyniku rozkaz LDA $2DA5.

Nieco więcej zmian trzeba wykonać w programie **Gyruss**. Liczba żyć jest tam zapisywana dwukrotnie. Trzeba więc odszukać rozkazy DEC $F7,X i DEC $AA,X oraz zamienić je na LDA $F7,X i LDA $AA,X. Liczbowe wartości tych rozkazów są następujące: $D6, $F7 (214, 247) zamienić na $B5, $F7 (181, 247) oraz $D6, $AA (214, 170) na $B5, $AA (181, 170). Przy operowaniu znakami ASCII trzeba w ciągach „V" w negatywie i „w" w negatywie oraz „V" w negatywie i „*" w negatywie zamienić pierwszy znak na „5" w negatywie.

Teraz duża poprawka dla posiadaczy stacji dysków. Dotyczy ono znanej gry **World Karate Championship**. Trzeba w niej odszukać blok rozkazów:

```
LDA  $D9
CMP  #$04
BCS  +$09
LDA  $DA
CMP  #$04
BCS  +$03
JMP  $01CF
```

Jest to ciąg liczb szesnastkowych: A5, D4, C9, 04, B0, 09, A5, DA, C9, 04, B0, 03, 4C, CF, 01. W tej sekwencji dokonuje się rozstrzygnięcie wyniku walki. Zamieńmy teraz te rozkazy na:

```
LDA  $DA
STA  $D9
LDA  $D9
CMP  #$01
NOP
NOP
BCS  +$03
JMP  $01CF
```

Powinniśmy uzyskać ciąg liczb: A5, DA, 85, D9, A5, D9, C9, 01, EA, EA, B0, 03, 4C, CF, 01. Po przystąpieniu do walki okaże się, że sędzia oślepł! Niezależnie od tego, co dzieje się na planszy, za każdym razem ogłasza zwycięstwo gracza białego. Oprócz tego wystarczy jedno zwycięstwo, aby przejść do następnej rundy.

Tomasz Wiśniewski
Wojciech Zientara

FLAGA PIRACKA ZNÓW NA MASZT!!!

W niemal każdym czasopiśmie komputerowym, zarówno w Polsce jak i na Zachodzie poruszany jest problem ochrony oprogramowania przed niedozwolonym kopiowaniem, czyli przed kradzieżą. Stan prawny istniejący w Polsce spowodował, że jedynym ewentualnym problemem pozostaje kupno komputera, natomiast oprogramowanie prawie samo wciska się nam w ręce.

Taki stan rzeczy spowodował, że nie doczekaliśmy się wielu udanych programów, które pisane byłyby w Polsce na potrzeby naszego skądinąd specyficznego rynku. Ten artykuł nie ma na celu odwoływania się do sumienia współrodaków o zaniechanie kopiowania programów (nazwijmy to otwarcie: kradzieży programów). Wiadomo, że dopóki nie zaistnieje powszechna kultura informatyczna, dopóki pozostanie taki stan prawny, a ewentualne nowe zarządzenia nie będą rygorystycznie przestrzegane, będziemy mieli w kraju tylko garstkę uczciwych informatyków, którzy oprogramowanie kupili lub stworzyli i z obrzydzeniem patrzą na wszystkie giełdy komputerowe, budy wypożyczające i handlujące nielegalnie posiadanym oprogramowaniem.

Należy od razu wspomnieć, że polscy „hackerzy" prawie nie istnieją — większość oprogramowania bowiem przybywa do nas z Holandii czy RFN już po odpowiednich przeróbkach i polskim pseudopiratom pozostaje jedynie „uzupełnienie" napisu firmowego lub podszycie się pod wcześniejszego pirata. Dlatego też można spokojnie mówić w Polsce o paserach i złodziejach.

Proceder podszywania się jest u nas wyjątkowo dobrze rozwinięty. Przez podszywanie rozumiem tu zmiany w nagłówkach programów mówiące np., że XXXX-Soft z Polski napisał program wspomagający proces projektowania samochodów FIAT (nie mylić z FSO na Żeraniu).

Artykuł ten ma na celu przybliżenie Czytelnikom kilku metod ochrony oprogramowania. Oczywiście opisane będą te z nich, które są już dość rozpowszechnione. Nowe supermetody zabezpieczeń ze względów bezpieczeństwa dla tych kilku programów, które nie zostały do tej pory „złamane", zostaną przemilczane.

Na początku należy odróżnić różne systemy zabezpieczania programów stosowane przez firmy software'owe i ochronę, przed kimś niepowołanym, naszego własnego oprogramowania.

W pierwszym przypadku system zabezpieczeń może być programowy (zależy wówczas od nośnika na jakim rozprowadzany jest program) lub sprzętowy. Zacznijmy od magnetofonu. Na początek idą programy, które automatycznie się uruchamiają. **Należy od razu powiedzieć, że system zabezpieczeń taśm kasetowych nie zdał absolutnie egzaminu w dobie magnetofonów dwukieszeniowych — daje to prawie idealne możliwości skopiowania dowolnego programu nagranego na taśmie. Ta metoda nie jest może powszechnie stosowana, ale jest popularna i znana.**

Samouruchamianie się programu w takim wypadku przestaje być jakimkolwiek problemem. Narzędziem coraz powszechniej stosowanym (do Commodore znane już było w 1981 roku, a więc w chwili gdy na rynku pojawił się VIC-20) jest specjalny moduł umożliwiający przyłączenie do komputera dwóch lub więcej magnetofonów — rola komputera ogranicza się jedynie do... ich zasilania. Komputer w żadnym momencie nie jest pytany czy operuje zabezpieczonym programem, nie odgrywa on w tym procesie żadnej roli.

Jako antidotum usiłowano swego czasu operować bardzo silnym sygnałem poczas nagrywania nawet do prze- sterowania włącznie. Metoda ta daje dobre rezultaty przede wszystkim na magnetofonach średniej jakości (układy nie są w stanie przenieść całego pasma co prowadzi do błędu odczytu przy wgrywaniu kopii programu do komputera). Innym sposobem jest zmiana prędkości obrotowej silnika magnetofonu dzięki czemu część programu wczytuje się szybciej a określone jego fragmenty wolniej. **Dodajmy, że my także mamy taką możliwość zmiany tej prędkości — odtworzenie tak zapisanego programu będzie wymagało odpowiednio działającego szyfru, który można zlokalizować np. w nagłówku programu. Podczas odtwarzania kopii pirackiej w większości wypadków otrzymamy LOAD ERROR.**

Magnetofon jednak przestał się już tak naprawdę liczyć (wybaczcie Ci wszyscy, którzy macie tylko magnetofon) w momencie, gdy spadły ceny na stacje dysków. Tu również istnieje szereg ciekawych sposobów zabezpieczania programów. Firmy stosują tu metodę błędnych sektorów. Są to takie sektory, w których specjalnie suma kontrolna nie odpowiada danym zapisanym w tym sektorze. W dobie programów kopiujących bit po bicie (nibblery) metoda ta straciła na znaczeniu gdyż jsteśmy w stanie otrzymać dokładną kopię wraz z błędami istniejącymi na dysku oryginalnym

Firmy dysponujące drogą aparaturą mogą również wytwarzać takie błędy mechanicznie, np. przez wypalenie dziurki laserem gdzieś na dysku. Tak wytworzony błąd jest niepowtarzalny na seryjnie produkowanych stacjach dysków. Znana jest także metoda dodatkowych sektorów i ścieżek. Szczególnie ta ostatnia była stosowana np. w oprogramowaniu na C-64 i stację 1541. Przy takiej metodzie pewne informacje były zapisywane na ścieżkach „nielegalnych" — 36-40. Programy kopiujące kopiowały cały dysk bit po bicie ale wyłącznie do ścieżki o numerze 35. Uruchomiony potem program szukał jakiejś informacji na ścieżkach 36-40 i nie zna|du|ąc jej (gdyż nie została ona skopiowana z resztą programu) stwierdzał, że nie ma do czynienia z-kopią i np. włączał procedurę formatującą dyskietkę.

Jak na taśmie, tak i w stacji dysków, przez wstawienie odpowiedniej wartości do komórki odpowiedzialnej za prę- dość nagrywania możemy utrudnić odtworzenie programu. **Należy dodać, że większa liczba zabezpieczeń, oprócz malejącego prawdopodobieństwa ich złamania, często odstrasza włamywaczy ze względu na ilość czasu koniecznego na stworzenie w pełni wartościowej niezabezpieczonej kopii. Często można spotkać kopie nie realizujące pewnych funkcji, co jest rezultatem właśnie włamań lub pośpiechu podczas odbezpieczania programu.** Wśród tych, których nie odstrasza ilość zabezpieczeń pozostają ewentualnie tylko fanatycy, którym akurat bardzo zależy na danym programie lub ci, którzy włamują się z czysto ambicjonalnych powodów. Generalnie jednak producenci opierają się obecnie przede wszystkim na błędach niemożliwych do skopiowania na seryjnych stacjach dysków.

Na koniec zostawiłem wszelkiego rodzaju przystawki sprzętowe utrudniające a nawet uniemożliwiające kopiowanie. Uniemożliwienie oznacza nieopłacalność stworzenia kopii takiej przystawki wobec np. niskiej ceny oryginalnego oprogramowania. Stosowane są więc różnego rodzaju nakładki na porty joysticków lub wyjścia magetofonu. Nakładka taka w postaci zalanej tworzywem sztucznym wtyczki, nie jest do niczego innego poza tym podłączona. Zawiera ona czasami jakieś informacje niezbędne do poprawnego działania programu, które są z niej przepisywane. Czasami program sprawdza tylko fizyczne istnienie wtyczki.

Popularne moduły (cartridge) produkowane są obecnie w wersjach z napylanymi na płytkę układami scalonymi. Często samo opakowanie modułu posiada przeprowadzoną pajęczą sieć cienkich połączeń. Otwarcie takiego pudełka oprócz niemożliwości skopiowania układów scalonych (EPROM) powoduje, że połączenia zostają przerwane a sam cartridge nadaje się bardziej na śmietnik niż do naszego komputera.

Na koniec kilka sposobików dla tych, którzy spośród już opisanych metod ochrony programów nie wybrali niczego dla siebie. Dobrym zabezpieczeniem jest skompilowanie programu i to najlepiej takim kompilatorem, do którego nie ma na rynku dekompilatora. Zawarcie w tak przygotowanym programie kilku pytań o hasło (najlepiej numeryczne, gdyż trudniej się je odszukuje) wyśmienicie utrudnia uruchomienie programu mimo całkowitej swobody w jego kopiowaniu.

Na rynku istnieją programy o nazwach Cruncher i Decruncher. Są to na ogół programy dostosowane do BASICa choć spotkałem się z odpowiednikiem tych programów operujących na wersji źródłowej programów pisanych w Pascalu. Programy te tworzą z naszej wersji programu (pisanej superstrukturalnie) stek pomieszanych pojęć, z których nie jesteśmy w stanie często wywnioskować w jakim języku pisany był program, nie mówiąc nic o jakichkolwiek zmianach w takim programie.

Ostatnio niektórzy producenci wpadli na jeszcze prostszy pomysł. **Nowopowstały program jest kierowany do sklepów bez jakichkolwiek zabezpieczeń, a więc i cena jego jest bardzo niska. Jeżeli zatem użytkownik może nabyć oryginalny program z instrukcją za, dajmy na to 10 dolarów, nie kupi on kopii bez instrukcji nawet na pół ceny — po prostu w tej sytuacji piractwo przestaje się opłacać.**

Dodajmy też, że oficjalny zakup ma wiele innych zalet. Gdy po zakupie wyślą swoją kartę rejestracyjną do producenta, ten ostatni zapewnia mu także, w wielu wypadkach, serwis dodatkowy. Przykładowo, gdy ukaże się na rynku następna wersja kupionego przeze mnie programu to jako prawowity i zarejestrowany użytkownik dostaję ją za śmieszną sumę kilku dolarów tylko dlatego, że byłem uczciwy.

Na całym świecie program komputerowy traktowany jest znacznie lepiej od samego komputera, gdyż jego ułożenie wymaga niewspółmiernie większego wysiłku twórczego. Jak wiadomo istnieją już fabryki automatyczne, gdzie człowieka się nie uświadczy; jak do tej pory nie ma jednak laboratoriów, w których wataha komputerów układałaby programy. W tej ostatniej produkcji człowiek jest nadal niezastąpiony, o czym nasze urzędy zajmujące się ponoć ochroną ludzkiej twórczości nie chcą absolutnie nic wiedzieć.

Dominik Falkowski



SEJF W PROGRAMIE

Artur Conan Doyle włożył w usta Sherlocka Holmesa powiedzenie: Co jeden człowiek zakrył drugi jest w stanie odtworzyć. O tym, że jest to najszczersza prawda nie potrzebuję chyba Czytelników przekonywać.

Istnieje wiele sposobów zabezpieczania programów przed kopiowaniem jak też i przed wścibskim okiem — wbrew pozorom jest to duża różnica. Zabezpieczenie przed kopiowaniem ma na celu maksymalne ograniczenie rozprowadzania programu dalej, natomiast zabezpieczenie przed podejrzeniem pozwala co prawda (choć nie zawsze) rozprowadzać program, jednakże bez możliwości ściągnięcia co lepszych pomysłów jego autora. W polskim Eldorado dla piratów (?) praktyczniejszy okazuje się ten drugi sposób.

Najbardziej bezpiecznym rozwiązaniem jest takie hasło, które po wprowadzeniu i sprawdzeniu samoistnie się skasuje, uniemożliwiając tym samym jego odczytanie. Hasłem może też być np. fragment programu, adres procedury napisanej w języku maszynowym itp. Załóżmy, że napisałeś program, który chcesz zabezpieczyć. Jako hasło wykorzystamy tu 5 wartości zapisanych w komórkach pamięci RAM.

Pierwszym krokiem, jaki powinieneś wykonać jest ZAWSZE kompilacja programu. Do jego wersji źródłowej wpisz inne proste zabezpieczenia np. POKE 792,193 (uniemożliwia wykonanie kombinacji STOP i RESTORE w C-64), czy pętlę testującą pewien obszar pamięci, np. na zawartość procedury w języku maszynowym. Po kompilacji wybieramy sobie fragment pamięci (niech w naszym przykładzie będą to komórki od 5000 do 5004) i za pomocą pętli w trybie ekranowym odczytujemy ich zawartość, zapisując je gdzieś w dobrze strzeżonym miejscu:

```
FOR X = 5000 TO 5004:PRINT PEEK(X);:NEXT
32 147 208 165 37
READY.
```

Za pomocą podobnej pętli (PRINT PEEK (X) zamieniamy na (POKE) wpisujemy teraz np. te same wartości zmniejszone o 2 czyli 30, 145, 206, 163, 35. Następnie zapisujemy program na taśmie czy dyskietce i... mamy problem z głowy. Podczas uruchamiania programu zgłosi on błędy wyglądające na wszystko tylko nie na hasło — włamywacz będzie raczej pewien, że problem leży w złym skopiowaniu czy oczywistych błędach popełnionych przed kompilacją. W takim wypadku nie pomoże nawet dekompilator, gdyż błędy są nadal; jeżeli teraz zabezpieczymy w ten sposób nie jeden ale kilka obszarów pamięci, to program stanie się nie do ruszenia...

Warto także zwrócić uwagę na fakt, że hasło należy w tym wypadku wpisać PRZED uruchomieniem programu (wpisując do komórek pamięci 5000—5004 właściwe wartości). W większości wypadków hasło podaje się zwykle PO uruchomieniu stąd też program sprawi włamywaczom więcej kłopotu.

Oczywiście oprócz tego możemy zastawić jeszcze inne pułapki na potencjalnego rzezimieszka. W komputerze mamy 65536 komórek pamięci, prawda? Jeżeli teraz program będzie sprawdzał przed uruchomieniem (lub cyklicznie) kilka określonych komórek pamięci, to nasze zabezpieczenie będzie pewniejsze.

Aby nie wskazywać włamywaczowi linii pułapek możesz zamiast SYS 64738 używać np. POKE 788, X czy POKE 789, PEEK(789)AND... Trik polega tu na wpisywaniu wartości do obszaru pamięci RAM, gdzie zawarte są wektory systemu operacyjnego; w obszarze tym wystarczy jeden nieprzemyślany ruch, aby zablokować komputer definitywnie.

Innym sposobem zabezpieczenia programu może być odczyt pierwszego wciśniętego klawisza czy odczytanie kodu umieszczonego na ekranie przed uruchomieniem programu. Wyobraźmy sobie taką procedurę;

Po włączeniu komputera wczytujemy program do pamięci. Następnie w ściśle określonym przez nas miejscu ekranu wpisujemy szereg znaków, z których składa się nasze hasło. Za pomocą RUN uruchamiamy skompilowany wcześniej program, który wpisuje w pewien obszar pamięci krótką procedurę w języku maszynowym. Jej zadaniem jest odczytanie i dodanie do siebie wartości dziesiętnych wszystkich znaków hasła. Stosujemy przy tym kody ekranowe tzn. litera A ma wartość 1, B — 2, odstęp — 32 itd. Przed rozpoczęciem zliczania do określonej komórki wpisujemy wartość sumy kontrolnej hasła. Jeżeli liczba ta będzie błędna komputer automatycznie zostanie wyzerowany.

W naszym przykładzie obszarer, jaki będziemy wisywać hasło będą komórki o adresie 1024, a ilość znaków wynosi 10. Pamiętajmy, że obszar, w który będziemy omawianą procedurę wpisywać MUSI być zerowany także — w przeciwnym wypadku włamywacz będzie miał ułatwione zadanie.

Program przykładowy opracowano dla C-64 i znajduje się on w obszarze od adresu 680 do 701 (jest to obszar zerowany podczas procedury inicjalizacyjnej — COLD START). Najpierw wpisz program oznaczony jako B. Przed jego uruchomieniem wpisz swoje .hasło od lewego rogu ekranu poczynając — potrzebnych jest 10 znaków. Po ich wpisaniu wykonaj; SYS 680: PRINT PEEK (2)

Otrzymaną wartość zanotuj — będziesz musiał wpisywać ją każdorazowo do komórki 693 przed uruchomieniem programu. Innymi słowy zanim wykonasz RUN musisz najpierw wykonać: POKE 693,20 Teraz wpisz hasło od lewego rogu ekranu poczynając i uruchom program. Każdy błąd będzie Cię kosztował powtarzanie całej procedury od nowa. PROCEDURA A:

```
680 CLC
681 LDX#0
683 TXA
684 ADC 1024,X
; ilość liter w haśle
; test sumy hasła w komórce 693
; hasło w porządku
; włamywacz — wyzeruj
687 INX
688 CPX#10
690 BNE 684
692 CMP+0

694 BNE 697
696 RTS
697 JSR 64738
PROCEDURA B:
680 CLC
681 LDX#0
683 TXA
684 ADC 1024,X
687 INX
688 CPX#10 ; odczytano wszystkie znaki (10)?
690 BNE 684 ; za mało obiegów,powtórz pętlę
692 STA 2 ; zapisz w komórce 2 sumę kontrolną hasła.
694 RTS
```

Dodajmy, iż pożądane byłoby, aby z kolei inna pętla zawarta w programie testowała obszar pamięci, gdzie znajduje się nasza procedura A, czy nie została ona przez jakiegoś spryciarza usunięta. W programie skompilowanym taki teścik jest w zasadzie nie do wykrycia.

Innym sposobem może też być kontrolowanie z jakiego urządzenia peryferyjnego korzystał użytkownik wczytując program do pamięci komputera. Do tego celu warto wykorzystać komórkę 186 zawierającą informacje o ostatnio używanym urządzeniu:

```
LDA 186
CMP#9      ;9 — numer urządzenia 9
BNE PIRAT
RTS
PIRAT JMP 64738
```

Ten przykład może być wyjątkowo złośliwy, wymaga bowiem od użytkownika zmiany numeru urządzenia stacji dysków PRZED wczytaniem programu. W każdym innym wypadku nastąpi wyzerowanie komputera.

Nieco inaczej wygląda sprawa z zabezpieczeniem programów napisanych w języku maszynowym. Tu ratunkiem pozostają nieopublikowane instrukcje mikroprocesora lub też duża ilość funkcji logicznych, takich jak AND, ORA, EOR czy BIT. Dobre wyniki daje też ukrywanie procedur czy fragmentów programów w obszarach zarezerwowanych dla ROM. Tak jak i w BASIC warto pozostawiać luki w programach, w które konkretne i właściwe wartości wpisujemy dopiero w momencie rozpoczęcia pracy z programem. Doskonałe wyniki daje maskowanie za pomocą EOR i wybranej komórki np. z ROM komputera:

```
         LDX#0                    LDX#0
LOOP     LDA49152,X      LOOP     LDA49152,X
         EOR 43215                EOR 43215,X
         STA49152,X               STA 49152,X
         INX                      INX
         BNE LOOP                 BNE LOOP
         RTS                      RTS
```

W podanym przykładzie kod maszynowy zawarty w komórkach od 49152 do 49152+255 został poddany operacji logicznej EXCLUSIVE OR z wartością stałą zapisaną w komórce pamięci ROM — 43215. Gdy po operacji maskowania (program może być używany zarówno do kodowania jak i dekodowania) obejrzysz ten fragment pamięci, przekonasz się, że nie odczyta tego bełkotu żaden disassembler. Zabezpiecz także adres komórki ROM (43215) lokalizując go w procedurze i wpisując tam (przed zapisaniem!) zera. Poprawne działanie tej procedury otrzymasz po uprzednim wpisaniu właściwego adresu w sposób podobny jak wpisywanie hasła w BASIC. Druga wersja tego programu poddaje operacji EOR 255 komórek RAM i 255 komórek ROM. Do wyboru do koloru...

Podane tu sposobiki mają na celu podsunięcie Czytelnikom pewnych rozwiązań, jakie mogą sami z łatwością zastosować w swoich programach. Ważne jest przy tym, aby pamiętać, że najlepiej program będzie chroniony jeżeli zastosujemy co najmniej dwa lub trzy z nich jednocześnie; w wypadku BASIC program powinien być też skompilowany. Najlepszą tarczą jest jednak zasada, że cennych programów nie należy udostępniać nikomu — nie ma lepszego zabezpieczenia niż własna nieufność...

Ze zrozumiałych względów nie byłem w stanie omówić innych, bardziej niezawodnych i skomplikowanych technik — wymagają one dobrej znajomości systemu operacyjnego i umiejętności programowania w języku maszynowym. Do tematu tego będziemy w Klanie Commodore co jakiś czas powracać; a może Czytelnicy zaproponują coś ekstra?

Na zakończenie mała niespodzianka. Jeżeli w C—64 wpiszemy w następujące komórki podane przy nich wartości, to nasz program stanie się niewrażliwy na SYS 64738, czy wciśnięcie jakiegokolwiek przycisku RESET. Na czym polega sztuczka wytłumaczę niebawem.

```
32772 — 195
32773 — 194
32774 — 205
32775 — 56
32776 — 48
```

Klaudiusz Dybowski

FINAL BASIC

C-64 doczekał się opracowania wielu rozszerzeń języka BASIC, żadne z nich jednak nie było tak uniwersalne jak to, które mamy przyjemność zaprezentować dziś Czytelnikom.
Wymagane wyposażenie:
Interfejs do wyobraźni
1.64773883902 KB poczucia humoru

Wychodząc naprzeciw potrzebom Czytelników KLAN COMMODORE uważnie śledzi prasę światową i bieżące trendy w informatyce. Nie mogliśmy zatem przeoczyć tak doskonałej wersji BASIC dla wszystkich komputerów Commodore jaką jest właśnie FINAŁ BASIC. Jej autor, Larry Cotton starał się zadowolić wszystkich entuzjastów Commodore i naszym zdaniem udało mu się to znakomicie. Nowe polecenia i instrukcje podane poniżej nie tylko rozszerzają BASIC V2.0, V3.5 i V7.0 ale dają użytkownikom możliwości nieograniczone. A oto pełna lista poleceń (formy podane w nawiasach oddają znaczenie polskie):

AND/OR (I/LUB):
Instrukcja logiczna. Jeżeli X jest prawdziwe AND/OR Y jest prawdziwe, wtedy Z będzie prawdopodobnie też prawdziwe ale niekoniecznie.

BUT (ALE):
Wyrażenie z logiki argumentatywnej. Przykład: A BUT B POSS C oznacza, że jeżeli A ALE NIE B jest fałszywe lub prawdopodobnie prawdziwe wtedy C będzie definitywnie, ale niekoniecznie większe niż suma A i B. Często używane z instrukcjami **WHY** (DLACZEGO) i **HOW** (JAK), jak też w postaci **BUT WHY i BUT HOW** (ALE DLACZEGO I ALE JAK)

DUMP():
Umożliwia przeniesienie zawartości obszaru pamięci określonego w nawiasie do obszaru absolutnie nieprzewidzianego.

EMU (xxxx):
Dozwolone tylko w trybie ekranowym. Powoduje przeistoczenie dowolnego Commodore w emulator dowolnego urządzenia kuchenno-domowego. Dozwolone parametry: **BLDR** (pralka), **MCWY** (kuchenka mikrofalowa), **RFGR** (lodówka), **STOY** (piec), **TLPH** (telefon), **TSTR** (toster).

EYER/BETTER(KIEDYKOLWIEK/LEPIEJ):
Używane z IF...THEN. Przykład: IF A EYER = 5 THEN B BETTER = 12.

GIGA:
Dodaje 1 gigabajt pamięci RAM do już istniejącej w dowolnym typie komputera. Uwaga: komputer może wymagać dodatkowego ekranowania i zmiany okablowania sieciowego w ścianach. Przed wykonaniem tej instrukcji załóż okulary ochronne i podłącz się osobiście do uziemienia.

GO (xxx):
Rozkaz podobny do GO64 w C-128. Jego wykonanie automatycznie zamieni Twój komputer w emulator innego komputera podanego w nawiasie. Dozwolone parametry: **IBM, APL, ATR, TRS i YIC HOW (JAK):**
Polecenie używane bardzo często przez początkujących.
Patrz także **WHY** (DLACZEGO)

HURRYUP:
Przyspiesza pracę stacji dysków 12 razy. Instrukcja kompatybilna ze wszystkimi rozkazami dyskowymi. Przykład: LOAD „tytuł",8,HURRYUP. Dodatkowo eliminuje stuki (głowicy i w głowie).

KOP:
Szybszy odpowiednik POKE, ponadto oszczędza Ci wypisywania jednej litery.

MAYBE (BYĆ MOŻE):
Instrukcja podobna do LET ale dająca znacznie więcej opcji. Przykład: IF X AND/OR Y = Z THEN MAYBE A = B: IF NOT THEN WHY NOT.

PLS (BŁAGAM):
Instrukcja której wykonanie powinno czasami poprzedzać RUN.

POSS:
Unikaj korzystania z tej instrukcji. Ze względu na błąd w ROM komputera POSS niszczy dokumentnie cały komputer.

READMAG (WCZYTAJ PROGRAM Z CZASOPISMA):
Umożliwia wczytanie programu podanego w dowolnym czasopiśmie komputerowym bezpośrednio do pamięci komputera. Wymaga złożenia strony z programem w kwadrat o boku 5.25" (lub 3.5" jeżeli masz stację 1581).

SHUFFLE (PRZETASUJ):
Najlepiej opracowana instrukcja FINAL BASIC i zarazem najszybszy algorytm desortujący dane. Przetasowuje losowo wcześniej posortowaną macierz o 65536 elementach w czasie mniejszym niż 4 nanosekundy.

SIT (SIADAJ):
instrukcja możliwa do wykonania tylko w trybie natychmiastowym. Czasami używana z **FETCH (PRZYNIEŚ), LIE DOWN (POŁÓŻ SIĘ) i SPEAK (MÓW).**

WHY (DLACZEGO):
Instrukcja uniwersalna. Może być stosowana sama jak np. **HOW (JAK)** lub też w kombinacjach z innymi instrukcjami logicznymi **NOT** (np. **IF NOT THEN WHY NOT**).

INSTRUKCJE GRAFICZNE I MUZYCZNE

BWCOL:
Umożliwia przeniesienie zawartości ekranu na drukarkę. Drukuje dowolny ekran (tekstowy lub graficzny) na czarnobiałej taśmie drukarki mozaikowej w pełnych kolorach. Papier sprzedawany jest oddzielnie

DUBL:
Podwaja rozdzielczość Twoiego komputera z 320x200 punktów do 640x200 punktów zwiększając jednocześnie ilość znaków w linii z 40 do 80. Wymaga ustawienia obok siebie dwóch monitorów (dokładnie bok w bok).

LISTEN:
Pozwala na rozpoznawanie poleceń wydawanych głosem. Wymaga zastosowania interfejsu Kurzweila, dwóch mikrofonów pojemnościowych firmy SHURE oraz przedwzmacniacza o mocy 2000 watów.

SCRIB:
Uniwersalna instrukcja umożliwiająca pisanie po ekranie czymkolwiek (pędzelkiem, ołówkiem, piórem wiecznym, różdżką czarodziejską). Interfejs sprzedawany oddzielnie.

YCRTY:
Poprzez syntezę fotoelektryczną, wybuchy supernowych i inne równie często występujące zjawiska, polepsza rozdzielczość komputera do rozdzielczości odbiornika telewizyjnego. Musi być używana w połączeniu z GIGA, DUBL i MORCOL.

Na podstawie COMPUTE!'s GAZETTE 2/88 opracował Klaudiusz Dybowski

OD REDAKCJI
W trakcie przygotowywania materiału do druku postanowiliśmy ulepszyć nieco FINAŁ BACIC dodając następujące instrukcje:

PERYF:
Można stosować tylko w programie. Umożliwia emulację stacji dysków 1541 po przyłączeniu do komputera adaptera BAMBINO 2 lub zastępuje DATASSETTE po włożeniu do portu magnetofonu zwykłej kasety C-60.

WSAD (n...):
Wyłącznie z drukarką. Pozwala na wydrukowanie banknotu o odpowiednim nominale niezbędnego nam na zakup dodatkowych urządzeń do naszego komputera. W miejsce kropek należy wpisać sumę (do 65536), w miejsce n symbol waluty.

ARGUM:
Dostarcza użytkownikowi niezbitych argumentów, że jego komputer jest znacznie lepszy niż kolegi. Dopuszcza się stosowanie w dyskusjach małżeńskich.

HIGH SCORE:
Instrukcja logiczna stosowana w grach wraz z IF...THEN. Zapewnia najwyższy wynik w dowolnej grze komputerowej. Stosować tylko w trybie ekranowym. Przykład IF JA THEN HIGH SCORE : IF KOLEGA THEN ZERO.

DO...DO:
Odmiana pętli FOR... NEXT. Umożliwia stosowanie czasoprzestrzeni, np. OD TERAZ DO TAMTEGO DRZEWA.

DREAM (...):
Instrukcja najbardziej uniwersalna — w miejsce kropek należy wpisać marzenie do realizacji. Zostanie ono przeniesione telepatycznie do komputera i wykonane w czasie krótszym niż 5 lat. Zastępuje także każdą wymaganą (i wymarzoną) w danym momencie instrukcję BASIC. Wpisanie złych życzeń pod czyimś adresem spowoduje wyświetlenie komunikatu ?DREAM ERROR.

KUKUŁKA:
Instrukcja przydatna podczas usuwania błędów z wpisanego dopiero co z czasopisma programu. Pozwala na odnalezienie i odczytanie zarówno samego błędu, jak i intencji autora. Uwaga: KUKUŁKA musi być stosowana wraz z ARGUM co pozwala na niezbite udowodnienie, że błąd popełnił autor.

TRUP:
Rozkaz ten pozwala na podanie objawów wybranej przez użytkownika choroby, którą chcemy symulować u internisty w celu uzyskania zwolnienia L4. Zbyt częste stosowanie prowadzi do poważnych powikłań pourazowych.

ZEGAR:
Udowadnia niezbicie rodzicom, że czas jaki spędziliśmy przy komputerze zamiast przy lekcjach jest niczym w porównaniu do czasu trwania Kosmosu.

(kd)

PRZEDSTAWIAMY WARSAW BASIC(5)

WŁASNE ZNAKI

W C-64 wyprowadzeniem znaków na ekran steruje procesor obrazów VIC-II. Mikroprocesor komunikuje się z tym sterownikiem poprzez pamięć ekranu. Wzorce znaków pobieranych do wyświetlania znajdują się w pamięci ROM od $d000 do $dfff.

Procesor obrazów dzieli całą pamięć C-64 na cztery banki po 16 Kb każdy. Standardowe położenie pamięci ekranu w pierwszym banku powoduje automatyczne odwołanie się do generatora znaków w ROM w banku czwartym.

Jeżeli chcemy w programach używać własnych znaków, to możemy umieścić w RAM własny generator znaków i poinformować o jego położeniu procesor obrazu. Własny generator znaków i pamięć ekranu muszą leżeć w obrębie jednego banku. Wiele „programów" demonstrujących możliwości definiowania własnych znaków poprzestaje na skopiowaniu systemowego generatora znaków do pierwszego banku, gdzie można go dowolnie modyfikować. Ten sposób wykorzystują również niektóre programy użytkowe (np. procesory tekstu). Nie jest to sposób wygodny wtedy, gdy chcemy mieć własne znaki w interpreterze BASIC, gdyż znacznie uszczupla on pamięć dostępną na tekst programu i zmienne.

W związku z tym w Warsaw BASIC zdecydowaliśmy się umieścić pamięć ekranu i generator znaków w RAM banku czwartego. Jest to najefektywniejszy z możliwych sposobów rozwiązania problemu własnych znaków, gdyż wcale nie uszczupla pamięci operacyjnej BASIC, bowiem wykorzystuje RAM znajdujący się pod standardowym generatorem. Ta część pamięci na ogół pozostaje i tak niewykorzystana. Tekst, który zamierzamy wyświetlić na ekranie może być złożony z dowolnych znaków, zaprogramowanych przez użytkownika, które w postaci całych zbiorów mogą być rzecz jasna zapisane na taśmie lub dyskietce. Odpowiedni zestaw znaków może być w dowolnym miejscu programu zmieniony na inny. W Warsaw BASIC służą do tego specjalne instrukcje i procedury. Nie wymaga to zatem dziesiątków POKE i innych zabiegów. Wszystkie wcześniejsze zaprogramowane zestawy czy alfabety mogą być wyprowadzone zarówno na ekran jak i na drukarki (MPS-801/803, Star NL-10, SG-10/15, LC-10 itp.).

W poprzednim odcinku przedstawiliśmy program, który modyfikuje konfigurację systemu C-64. Jest ona zgodna z konfiguracją pamięci Warsaw BASIC.

W tym odcinku pokażemy zatem jak z tego korzystać do wyprowadzania własnych znaków na ekran monitora. Przedstawiony w poprzednim odcinku program kopiuje standardowy generator z ROM do RAM od $d000 do $dfff i przełącza procesor obrazu na drugi zbiór znaków (tryb biurowy — małe i duże litery). Oznacza to, że wyprowadzanie własnych znaków w naszym systemie polega na zmianie zbioru znaków trybu biurowego (2Kb od $d800 do $dfff).

Aby zmienić jeden znak należy: zaprojektować go w postaci tablicy o wymiarach 8x8, zakodować w postaci ciągu ośmiu liczb z zakresu od 0 do 255 i wstawić ten ciąg w odpowiednie miejsce do generatora w RAM.

W obszarze, którego zawartość chcemy zmieniać znajduje się pod tym adresem: rejestry układów we/wy, wspomniany wyżej standardowy generator znaków i pamięć RAM, w której jest już umieszczony generator własny. O tym, co widzi aktualnie mikroprocesor w tym obszarze decyduje zawartość rejestru $0001. Rejestr ten decyduje również o dostępności innych obszarów RAM, których adresy są takie same jak adresy ROM zawierającego standardowy interpreter BASIC i jądro systemu operacyjnego. W Warsaw BASIC dotyczy to również pamięci od $8000 do $9fff, wykorzystywanej przez ekspander.

Program prezentowany w poprzednim odcinku zawiera procedury, które pozwalają włączać dostęp do własnego generatora (sys 50440) i przywracać dostęp do rejestrów we/wy (sys 50423). instrukcje te mogą być używane w trybie programowym i po wywołaniu pierwszej z nich (sys 50440) instrukcja POKE może służyć do modyfikowania zawartości własnego generatora znaków. Efekty zmian dokonanych w zbiorze znaków można obserwować bezpośrednio na ekranie monitora. Po dokonaniu zmian w zbiorze znaków należy obowiązkowo przywrócić komunikację z rejestrami we/wy (sys 50423).

Na zakończenie przedstawiamy przykład modyfikacji znaku, którego kod ekranowy ma wartość PC (porównaj zamieszczony obok program). Projekt własnego znaku i sposób kodowania wyjaśniono na załączonym rysunku. Należy zauważyć, że punkty do zaświecenia są oznaczone gwiazdką, a miejsca puste wypełniają kropki. We wzorach kropkom odpowiadają zera, a gwiazdkom odpowiednie liczby. Ich wartość zależy od kolumny, w której występuje gwiazdka.

Przedstawiliśmy tutaj modyfikację pojedynczego znaku. Inne znaki można podobnie projektować i kodować.

Aby wymienić zbiory znaków automatycznie tak jak w Warsaw BASIC, przydałoby się móc w naszym Bajtkowym systemie definiować i wywoływać procedury. A o tym będziemy pisać w następnych odcinkach.

K. Gajewski
B. Radziszewski

```
10 x=12*4096+4*256+15*16+7
20 y=12*4096+5*256+ 0*16+8
30 sysy -       rem"RAM on"
40 pc=160:rem"kod ekranowy"
50 z=13*4096+8*256+pc*8
60 fori=0to7
70 reada:pokez+i,a
80 next
90 sysx :rem       "I/O on"
100 data0,138.137,170,217,138,0,0
ready.
```

```
84218421
........ (0+0+0+0)*16+0+0+0+0 =   0
*...*.*. (8+0+0+0)*16+8+0+2+0 = 138
*...*..* (8+0+0+0)*16+8+0+0+1 = 137
*.*.*.*. (8+0+2+0)*16+8+0+2+0 = 170
**.**..* (8+4+0+1)*16+8+0+0+1 = 217
*...*.*. (8+0+0+0)*16+8+0+2+0 = 138
........ (0+0+0+0)*16+0+0+0+0 =   0
........ (0+0+0+0)*16+0+0+0+0 =   0
```

LABIRYNT

Bajtkowe przedszkolaki z grupy Romka dotarły już do „starszaków" — czas najwyższy, by czytać kolumny „Klanów". Zanim więc rozpoczniecie naukę asemblera, uświadomcie sobie, że BASIC to nie tylko Kubuś Literka. Dla Was kontynuujemy cykl „Zrób to sam" — czyli układamy „duży" program w oparciu o pomysł z „Bajtka".

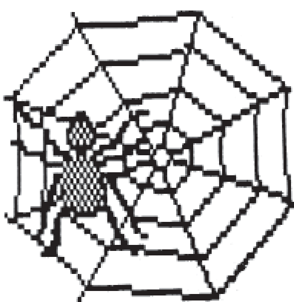
Dziś w KLANIE SPECTRUM propozycja zabawy w bieganie po labiryncie. W podanym przykładzie (jest to tylko przykład) należy w jak najkrótszym czasie przejść wszystkie korytarze. Operując czterema klawiszami (Q, A oraz O i P), zbieramy porozrzucane gwiazdki. Kto chce, może już zacząć grać. Należy tylko pamiętać, by po wpisaniu programu utrwalić go na taśmie oraz by w czasie gry nie był włączony CAPS LOCK. Tym zaś, którzy chcieliby tanim kosztem stworzyć „własną" grę, proponujemy analizę załączonego przykładu i co najważniejsze, udoskonalenie go, bo przecież nie możemy w „Bajtku" drukować super wersji na 48 kB.

Wszystkim radzimy najpierw wpisanie programu dokładnie w takiej postaci, jak na listingu. Pierwszą rzeczą jaką będziecie potem mogli zmienić jest kształt labiryntu, zapisany w liniach DATA. By nie pogubić się w zerach i jedynkach, dobrze byłoby najpierw zaprojektować go na kartce (labirynt zajmuje obszar 32 x 19 znaków). W liniach 2000 — 2030 zdefiniowana jest postać, którą poruszać się będziecie po labiryncie (aby ją „wywołać" przy wpisywaniu programu, należy najpierw wpisać i uruchomić fragment 2000—2030, a następnie używać GRAPHICS i A). Postać tę możecie również dowolnie zmieniać — sposób definiowania wyjaśniony jest w każdej instrukcji do komputera. Klawisze sterujące „schowane" są w liniach 90—120. Można je wymienić na inne lub, po dopisaniu kilku linii, wprowadzić możliwość definiowania ich przez gracza. Wiele linii można też dodać, by ulepszyć oprawę graficzną i dźwiękową. Pomysłów na pewno Wam nie zabraknie — wszak nie jesteście już przedszkolakami.

(Pb)

```
 10 BORDER 0: INK 7: PAPER 0: CLS
    : LET d=0: LET y=3: LET x=-1:
    LET r=1000
 15 GO SUB 2000: RESTORE
 20 FOR i=1 TO 608: READ a
 30 IF a=1 THEN PRINT AT
    y,x+1;"#": GO TO 50
 40 INK 5: PRINT AT y,x+1;"#": INK
    7
 50 LET x=x+1: IF x=31 THEN LET
    y=y+1: LET x=-1
 60 NEXT i: PRINT AT 1,20;"Rekord
    ";r: PRINT AT 3,13;" START ":
    PAUSE 0: BEEP .05,30: PRINT AT
    3,13;"#######"
 70 LET t=0: LET y=4: LET x=5: LET
    p=0
 80 PRINT AT y,x;"[graphic]"
 90 IF INKEY$="q" THEN GO SUB 500
100 IF INKEY$="a" THEN GO SUB 530
110 IF INKEY$="o" THEN GO SUB 560
120 IF INKEY$="p" THEN GO SUB 600
130 LET t=t+.5: IF p=304 THEN GO
    TO 200
140 PRINT AT 1,7;"Czas ";INT (t)
150 GO TO 85
200 BEEP .2,15: BEEP .2,11: PRINT
AT 1,7;"CZAS ";INT (t)
210 IF INT (t)<r THEN PRINT AT
    1,20;"Rekord ";INT (t): LET
    r=INT (t)
220 INPUT #0;"GRASZ JESZCZE RAZ ?
    (T)";o$
230 IF o$="t" THEN RESTORE : GO TO
    10
240 BEEP .3,10: STOP
```

```
 500 IF  SCREEN$ (y-1,x)="*" THEN
     LET p=p+1: LET y=y-1: PRINT AT
     y,x;"≛";AT y+1,x;" "
 510 IF  SCREEN$ (y-1,x)=" " THEN
     LET y=y-1: PRINT AT y,x;"≛";AT
     y+1,x;" "
 520 RETURN
 530 IF  SCREEN$ (y+1,x)="*" THEN
     LET p=p+1: LET y=y+1: PRINT AT
     y,x;"≛";AT y-1,x;" "
 540 IF  SCREEN$ (y+1,x)=" " THEN
     LET y=y+1: PRINT AT y,x;"≛";AT
     y-1,x;" "
 550 RETURN
 560 IF  SCREEN$ (y,x-1)="*" THEN
     LET p=p+1: LET x=x-1: PRINT AT
     y,x;"≛";AT y,x+1;" "
 570 IF  SCREEN$ (y,x-1)=" " THEN
     LET x=x-1: PRINT AT y,x;"≛";AT
     y,x+1;" "
 580 RETURN
 600 IF  SCREEN$ (y,x+1)="*" THEN
     LET p=p+1: LET x=x+1: PRINT AT
     y,x;"≛";AT y,x-1;" "
 610 IF  SCREEN$ (y,x+1)=" " THEN
     LET x=x+1: PRINT AT y,x;"≛";AT
     y,x-1;" "
 620 RETURN
1000 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1
     ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
     ,1,1,1,1: REM 32 jedynki
1010 DATA 1,0,1,0,0,0,0,0,0,0,0,0,0
     ,0,0,1,0,0,0,0,1,0,0,1,0,0,0,0
     ,0,0,0,1
1020 DATA 1,0,1,0,1,1,0,1,1,1,1,0,1
     ,1,0,1,0,1,1,1,0,0,1,0,0,1,0,1
     ,1,1,0,1
1030 DATA 1,0,0,0,1,1,0,0,1,0,0,0,0
     ,1,8,1,0,1,0,1,0,1,1,0,1,1,0,1
     ,0,0,0,1
1040 DATA 1,0,1,0,0,0,1,0,0,0,1,1,0
     ,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
     ,1,1,0,1
1050 DATA 1,0,1,1,1,0,0,0,1,0,0,1,0
     ,1,0,1,1,0,1,0,1,1,1,0,0,0,1,1
     ,1,0,0,1
1060 DATA 1,0,0,0,0,1,0,1,1,0,1,0,0
     ,0,1,0,0,0,1,0,1,1,0,1,0,1,1,1
     ,0,0,0,1
1070 DATA 1,0,1,0,0,0,0,0,0,0,0,1,0
     ,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
     ,1,0,0,1
1080 DATA 1,0,1,1,1,0,1,1,1,0,1,1,1
     ,1,0,0,0,1,1,0,1,0,1,0,1,1,1,0
     ,1,1,0,1
1090 DATA 1,0,1,0,0,0,1,0,1,0,0,1,0
     ,1,0,1,0,1,0,0,1,1,1,0,1,0,1,0
     ,1,0,0,1
1100 DATA 1,0,1,1,1,0,1,1,1,1,0,1,1
     ,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0
     ,1,0,1,1
1110 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0
     ,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0
     ,0,0,1,1
1120 DATA 1,0,1,1,1,0,1,1,1,1,1,0,1
     ,0,1,0,1,1,0,1,0,0,0,1,1,1,1,0
     ,1,0,1,1
1130 DATA 1,0,1,0,1,0,0,0,0,0,1,1,1
     ,1,1,0,1,1,0,1,1,1,0,0,0,1,0,0
     ,1,1,0,1
1140 DATA 1,0,0,0,1,1,1,0,1,0,1,0,0
     ,0,0,0,0,0,0,0,1,0,1,0,1,0,0,1
     ,0,0,0,1
1150 DATA 1,0,1,1,1,0,0,0,1,0,1,0,1
     ,0,1,1,1,0,1,1,1,0,0,0,0,1,0,0
     ,0,1,0,1
1160 DATA 1,0,1,0,0,0,1,1,1,0,0,0,1
     ,0,0,0,0,0,0,0,1,0,1,1,0,1,0,1
     ,1,1,0,1
1170 DATA 1,0,0,0,1,0,0,0,0,0,1,1,0
     ,0,1,1,0,1,1,0,0,0,1,0,0,0,0,0
     ,0,0,0,1
1180 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1
     ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
     ,1,1,1,1
2000 RESTORE 2030: FOR f=USR "a" TO
     USR "a"+7
2010 READ a: POKE f,a: NEXT f
2020 RETURN
2030 DATA
     24,60,24,255,24,126,66,195
```



POSŁUSZNE TABELE

Basic ZX Spectrum drukuje liczby z wyrównaniem do pierwszej kolumny. Przy liczbie o wartości bezwzględnej mniejszej od 0.1 opuszczone jest w wydruku zero.

Aby otrzymać czytelniejszy wydruk proponuję zastosować opcję TAB i kilka funkcji zdefiniowanych za pomocą polecenia DEF FN.

Argumentem opcji TAB jest **k-FN a(x)-1**, gdzie:

k — numer kolumny przeznaczonej dla kropki

a(x) — funkcja przyjmująca następujące wartości:

0 dla -0.1<x<0.1 z wyjątkiem x = 0, kiedy **FN a(0)**=1

1 dla 0.1 ≤ | x | <10

2 dla 10 ≤ | x | <100

3 dla 100 ≤ | x | <1000 itd.

— 1 — przesunięcie o 1 kolumnę w celu uzyskania pola na znak. Przy wydruku liczb nieujemnych można opuścić -1 oraz **FN b$(x)**

Konstrukcja funkcji **a(x)** jest następująca:

FN a(x) = składniki + składnik2 +składnik3

Składnik 1 ma przyjąć wartość 1 dla zera oraz zero dla pozostałych liczb.

składnik1 = 1 - ABS SGN x

Przy braku zera w danych przeznaczonych do drukowania można ten składnik pominąć.

Składnik 2 ma przyjąć wartość zero dla liczb większych od minus 0.1 i mniejszych od 0.1 (z wyjątkiem zera), zaś 1 dla pozostałych

składnik 2 = SGN INT ABS (10.000000016 ,x)*)

Składnik 3 ma przyjąć wartość zero dla liczb o wartościach bezwzględnych mniejszych od 10, 1 dla liczb od -100 do -10 i od 10 do 100, 2 dla liczb od -1000 do -100 i od 100 do 1000 itd.

składnik3 - SGN INT ABS(0.10000000018*x) + SGN INT ABS(0.010000000006*x) + ...

Liczba członów składnika 3 zależy od wielkości największych liczb, które będą drukowane w tabeli. Gdy w wydruku nie wystąpi zero, **składnik 3** może mieć postać:

składnik3 = INT((LN ABS x/LN 10)* czynnik)

Czynnik potrzebny jest, aby składnik 3 nie przyjmował wartości ujemnych dla | x | < 1.

czynnik (1 + SGN LN ABS x) * 0.50000000006

Gdy w wydruku występują liczby o wartości bezwzględnej nie mniejszej od 1 oraz liczba zero, to **LN ABS x** można zastąpić przez **VAL (CHR$(32 < (184-32)* ABS SGN x) + „ABS" + STRS x)**)**

Argumentami instrukcji PRINT są funkcje **b$(x)** oraz **c(x)**. Wartością funkcji **b$** jest znak minus dla liczby ujemnej oraz spacja dla liczby nieujemnej.

DEF FN b$(x) = CHR$ (45 13*SGN (1 + SGN(x))

Funkcja c określa dokładność wydruku, np. przy dwóch miejscach po kropce wygląda ona następująco:

DEF FN c(x) = ABS (0.01 * INT(100*x + 0.5))

Jeżeli w różnych kolumnach żądana jest różna liczba miejsc po kropce, to można zdefiniować następne funkcje **d(x)**, **e(x)**...

Przykładowa instrukcja wydruku wygląda więc następująco:

PRINT TAB k-FN a(FN c(x)) -1;FN b$(x); FN c(x); TAB 1 - FN a (FN c(y)) -1; FN b$(y); FN c(y);; TAB m FN a (FN d(z)) -1;FN b$(z);FN d(z)

Marek Nowosad

— wykonując obliczenia otrzymano I**NT 1 = 1**, ale **INT (10*0.1) =0,**

— dlatego zamiast 10 występuje 10.000000016. Podobnie w dalszych obliczeniach zamiast 0.1, 0,01 i 0.5 wprowadzono liczby nieco większe

— błąd w systemie operacyjnym powoduje, ze dla | x | <1 z wyjątkiem x=0 opuszczone zostają znaki przed funkcją STR$ x, np.: **ciąg znaków + STR$ x -+ ciąg znaków = STR$ x + ciąg znaków**

WERYFIKATOR

Jak sprawdzić, czy wpisany właśnie program jest poprawny?

Najlepiej porównać go z oryginałem. Może to być jednak niewykonalne. Proponuję metodę stosowaną już w Klanie Atari i Klanie Commodore. Sposób będzie jednak trochę inny. Spectrum posiada edytor na tyle nietypowy, że właściwie nie ma możliwości kontrolowania aktualnie wpisanej linii. Zastosowałem więc inną metodę — dopisywania do wprowadzonej już linii sumy wszystkich jej kodów modulo 255 w kodzie szesnastkowym.

Procedura pracuje w dwóch wersjach. Pierwsza, uruchamiana przez RANDOMIZE USR 23310 powoduje zapisanie sumy kontrolnej w każdej linii, którą kończy instrukcja REM xx. XX to dwa dowolne znaki.

Tryb uruchamiany przez RANDOMIZE USR 23380 powoduje dopisanie do każdej istniejącej linii końcówki REM z kodem kontrolnym linii.

Proponuję przeanalizowanie procedury w kodzie maszynowym. Jest ona umieszczana w buforze drukarki i pozostaje tam do chwili wyzerowania komputera. Oczywiście, jeżeli używa się drukarki, procedurę należy przenieść w inny obszar pamięci lub wpisywać ją tam po każdym użyciu drukarki.

Mieczysław Torbus

Od redakcji: Prosimy autorów piszących do Klanu Spectrum o nadsyłanie listingów potraktowanych przez powyższą procedurę. Pozwoli to na uniknięcie wielu błędów przy wpisywaniu programów.

```
10 FOR f=23296 TO 23474
20 READ a:POKE f,a
30 LET s=s+a
40 NEXT f
50 IF s<>16986 THEN PRINT
"Zle dane!"
1000 DATA 0,35,230,15,198,4
8,254,58,56,2,198,7,119,20
1,42,75,92,229,42,83,92,24,
4,35,35,35,35,35, 35,167,20
9,237,82,25,48,233,213,94,3
5,86,43,43,43,20,175,79,129
, 79,126,35,29,32,249,21,32
,246,126,254,234,32,218,121
,15,15,15,15,205,1,91,121,2
05,1,91,24,206,58,234,0,0,1
3,0,0,0,0,221,33,0,0,42,75,
92,43,235,42,83,92,213,205,
184,25,225,167,237,82,25,23
5,221,35,48,242,221,229,225
,36,34,176,92,37,17,4,0,205
,169,48,68,77,42,83,92,205,
85,22,221,33,176,92,19,35,2
35,35,35,78,35,70,3,3,3,3,1
12,43,113,43,43,237,176,229
,1,5,0,27,33,75,91,237,176,
225,221,53,0,32,224,221,53,
1,32,219,195,14,91
```

CPS 8256

CZYLI OKAZJA DO POCHWAŁY KOMPUTERA

O tym, że Amstrad PCW jest maszyną przeznaczoną do przetwarzania tekstów wiedzą wszyscy, zainteresowani sprzętem komputerowym. W takim charakterze komputer ten reklamowany jest na zachodzie Europy i w Stanach Zjednoczonych, gdzie cieszy się olbrzymią popularnością. Natomiast niewiele chyba osób zdaje sobie sprawę, z doskonałych możliwości Amstrada PCW przy pracy w systemie CP/M, a przecież jest to w dalszym ciągu jeden z najpopularniejszych systemów operacyjnych.

W naszych warunkach, gdy do poziomu pełnego nasycenia sprzętem komputerowym, porównywalnego z innymi krajami, bardziej rozwiniętymi, brakuje nam sporo, ten tryb pracy omawianego komputera powinien być bardzo istotny.

Przy nacisku położonym na profesjonalne wykorzył stanie Amstrada PCW, a nie na gry, maszyna ta zapewnia znacznie większe możliwości i wygodę niż cieszący się dużym powodzeniem model Amstrada CPC 6128. A przy tym PCW jest tańsze, niezależnie od tego czy bierzemy pod uwagę ceny w dolarach na Zachodzie, czy w złotówkach u pośredników. W jednej z firm wysyłkowych CPC 6128 plus drukarka DMP 2000 kosztuje niewiele mniej niż 400 funtów, natomiast PCW 8256 można nabyć za około 300. Oprócz tego należy uwzględnić dwukrotnie mniejszą pamięć CPC, której rozszerzenie będzie nas kosztować kilkadziesiąt funtów.

Dodatkową zaletą PCW jest także zawartość samego komputera i jego rozszerzeń. Przykładowo zwiększenie pamięci do 512 KB nie wymaga żadnych przystawek, wystarczy odkręcić 6 śrub obudowy monitora zobaczymy 8 pustych podstawek pod układy scalone 256 kbit. Wkładamy je, przestawiamy dwa mikroprzełączniki na głównej płycie i przykręciwszy ponownie 6 śrub mamy 512 KB RAM, z których 368 idzie na utworzenie tzw. Ramdysku, dostępnego nie tylko w systemie CP/M, ale również pod Locoscriptem.

Podobnie wygląda sprawa z drugą stacją dysków do tego komputera. Też nie umieszczamy jej na zewnątrz. W obudowie są gotowe wiązki przewodów zakończone wtykami, a samą stację mocujemy pod pierwszą stację w miejsce firmowej zaślepki. Trochę problemów mamy, jeśli chcemy dołączyć stację 5 i 1/4 cala, ponieważ system operacyjny PCW chciałby widzieć w tym miejscu dysk o podwójnej gęstości zapisu (80 ścieżek), taki jak w komputerze IBM PC/AT. Podłączenie typowej stacji 40-ścieżkowej możliwe jest po zmianie tablic parametrów dyskowych systemu operacyjnego.

Wiele osób twierdzi, że oryginalna drukarka Amstrada PCW nie wygląda solidnie. Oprócz mojej własnej, pozytywnej opinii, na podstawie ponad rocznej eksploatacji sprzętu, mogę pocieszyć malkontentów, także i w inny sposób. Za niecałe 50 funtów można kupić przystawkę CPS 8256, której umieszczenie na tylnej ściance monitora nie zwiększa istotnie jego gabarytów, ponieważ projektanci komputera przewidzieli miejsce na jej instalację. Wewnątrz znajdują się dwa interface'y: równoległy typu CENTRONICS i szeregowy RS- 232C. Należy żałować, że przystawka ta nie jest wliczona w podstawową cenę sprzętu, albowiem stanowi ona bardzo ważne uzupełnienie możliwości Amstrada PCW.

Interface równoległy służy do podłączenia dodatkowej drukarki lub plotera. Drukarka SG15 świetnie nadaje się do tego celu, ponieważ pozwala na stosowanie szerokiego papieru — 15 cali i ma kody sterujące kompatybilne z oryginalną drukarką. Z podłączeniem plotera też nie ma większych problemów. W razie kłopotów możemy skorzystać ze złącza szeregowego, które oprócz tego daje szereg ciekawych możliwości. Przyłączenie modemu jest jedną z nich, choć niestety w naszym kraju, gdzie trudniej o telefon niż o komputer, a jakość łączy przedstawia wiele do życzenia, nie jest to łatwe.

Realniejsze wydaje się wykorzystanie złącza RS 232 C do lokalnej komunikacji między komputerami (do 15 m), a w szczególności do pracy terminalowej. Amstrad PCW 8256 jest przygotowany do tego celu także od strony softwarowej. Program MAIL232.COM, który z powodu braku miejsca na CP/M-owej stronie dyskietki systemowej został umieszczony obok zbiorów Locoscripta, zapewnia emulację terminala typu WT52 i transfer zbiorów do i z komputera głównego. W przypadku tworzenia lokalnych sieci terminalowych, te możliwości PCW nie są do pogardzenia. Wszędzie, gdzie mamy sprzęt zakupiony przy okazji nabycia komputerów IBM, tzn. drukarki, plotery, dyski 5 i 1/4 cala do AT, Amstrad PCW może również korzystać z niego.

Obecnie nie ma także problemów z oprogramowaniem do PCW, co było początkowo dość istotnym ograniczeniem. Całe oprogramowanie dostępne pod CP/M Plus na CPC 6128, daje się zainstalować w nowym modelu Amstrada. Wordstar lub Newword są alternatywnymi edytorami tekstu dostępnymi pod CP/M, a sam Locoscript doczekał się już wielu, nie tylko polskich wcieleń. Dbase II, Multiplan, Supercalc II działają również dobrze jak na CPC, a dzięki dużemu RAM-dy- skowi, znacznie szybciej i wygodniej. Kompilator języka C firmy Hisoft, pozwala wykorzystać duże możliwości graficznego rozszerzenia do systemu CP/M, jakim jest GSX (Graphic System Extension).

Ekran o rozdzielczości 720 na 256 punktów lub 32 linie o 90 kolumnach pozwala zmieścić znacznie więcej informacji niż typowe monitory mające 24 linie po 80 kolumn. Turbo-Pascal jest jednym z moich ulubionych produktów software'owych, z jakich można korzystać na PCW. Pracuje się na nim niemal jak na IBM'ie, a wersję źródłową można uruchomić na dowolnym innym komputerze. Na Amstradzie dostępna jest także okrojona wersja programu Autocad, a para programów firmy Digital Research — DR Draw i Dr Graph, pozwala użytkownikowi na obejrzenie na ekranie własnych rysunków lub wykresów, a także na późniejsze ich wydrukowanie na drukarce lub ploterze.

Biorąc pod uwagę wyżej podane fakty, uważam, że Amstrad PCW 8256 jest najlepszym i najtańszym, dostępnym na naszym rynku, kompletnym zestawem do pracy pod kontrolą systemu operacyjnego CP/M. Z opinią tą niestety nie zgadzają się koledzy z sąsiednich stron „Bajtka".

(j.m.)

ROZWÓD W MIKROSKALI

Marcowe, międzynarodowe targi informatyczne CeBIT odbywające się w Hanowerze potwierdziły ostatecznie rozpad duetu Amstrad-Schneider. Kilkuletnia współpraca, rozpoczęta wytwarzaniem sprzętu Hi-Fi i mikrokomputerów domowych CPC 464 — 6128 zakończyła się dwoma mikrokomputerami osobistymi PC1512 i PC1640, oferowanymi nadal pod wspólną firmą, ale już na osobnych, własnych stoiskach.

Wydaje się, że Schneider ma duże szanse, by rozstanie obrócić w sukces rynkowy i techniczny. Świadczą o tym nowe modele komputerów osobistych, a mianowicie: Euro PC — dość tani (1300 DM) mikrokomputer z wyglądu przypominający Atari ST; PC XT typu „wieża"; przenośny PC o nazwie Target i znany już z wystawy w PKiN Schneider 2640 — odpowiednik IBM PC/AT.

W porównaniu ze Schneiderem Amstrad jak gdyby stracił rozmach: właściwie nie ma żadnych nowości, bo trudno do takich zaliczyć specjalizowany mikrokomputer do przetwarzania tekstów PCW 9512 (rozwinięcie modeli PCW 8512 i 8640), czy przenośny PPC 512/ 640 (zresztą w moim odczuciu pod wieloma względami nieudany). Prezentowany w lutym br. na Amstrad Computer Show nowy model domowego mikrokomputera 8-bitowego Spectrum +3 też budzi mieszane uczucia. Z jednej strony niejako dublujący CPC 6128 (128 KB RAM, 3" napęd dyskowy), a z drugiej strony kompatybilny ze Spectrum, wyjawia pewną słabość firmy: na własnym, angielskim rynku Amstrad nie cieszy się aż taką popularnością, jak np. we Francji.

W dodatku sir Sinclair nie rezygnuje i jego nowy „ręczny" mikrokomputer Z-88, będąc konstrukcją pośrednią między mikrokomputerami domowymi i osobistymi (do 3 MB pamięci RAM, możliwość wymiany zbiorów z PC, cena 250 funtów), może pokrzyżować Amstradowi szyki. Na razie jest jednak za wcześnie na wyrokowanie o przyszłych losach byłych partnerów. Dopiero czas pokaże kto zyskał: Amstrad czy Schneider?

W.Z.

COMHEX

-TROCHĘ INFORMACJI O TRANSFERACH ZŁĄCZEM RS-232C

Posiadacz komputera obdarzonego rzadkim typem formatu dyskietki ma często kłopoty z otrzymaniem nowych programów. Jednym z możliwych rozwiązań jest zastosowanie złącza RS-232C. Ale i w tym przypadku możemy mieć problem z oprogramowaniem komunikacyjnym. W systemie CP/M istnieje komenda PIPCOM, która umożliwia transfer zbiorów, ale niekiedy nałożone są ograniczenia na długość i format przesyłanego zbioru.

Z takimi właśnie problemami można zetknąć się podczas przenoszenia zbiorów typu COM w CP/M. Są to zbiory, w których wszystkie 8 bitów każdego słowa jest ważnych, a program PIP obsługujący interfejs RS-232C przesyła tylko 7 bitów. W niektórych komputerach (np. SuperBrain QD II) transmitowany w ten sposób zbiór nie może mieć więcej niż 25 KB.

Proponowane razem z programem COMHEX rozwiązanie polega na wykonaniu następujących operacji:

1. Przekształcenie zbioru typu COM np. X COM na kilka zbiorów typu HEX (XCOM, 001, XCOM, 002, itd.) przy pomocy programu COM- HEX.
2. Połączenie kablem i ustawienie tych samych parametrów transmisji złącz RS--232C (w systemie CP/M Plus 8-bitowych komputerów firmy Amstrad służy do tego komenda SETSIO.COM:

SETSIO TX 1200 RX 1200 BITS 7 PARITY EVEN STOP 1 NO HANDSHAKE, jest całkiem bezpieczna, tzn. zapewnia pewny transfer zbiorów przy dość dużej szybkości).

3. Użycie komendy PIP COM na obu komputerach (w systemie CP/M):
   a) pip XCOM,001 =aux; (odbiorca)
   b) pip aux; = XCOM,001
   pip aux; = eof; (nadawca)
4. Powtórzenie te] sekwencji dla zbiorów XCOM.002, XCOM.003 itd.
5. Ponowne zastosowanie komendy PIP u odbiorcy celem scalenia odebranych zbiorów;
   pipX,HEX;=XCOM,001,XCOM,002..........
6. Podziałanie na zbiór X,HEX komendą LOAD.COM (CP/M 2,2) lub HEXCOM.COM (CP/M Plus);
   load X,HEX

W wyniku tych operacji na komputerze odbierającym znajdzie się zbiór X.COM.

Ideą tego rozwiązania jest minimalizacja koniecznych narzędzi programistycznych na komputerze docelowym (PIP i LOAD). Cały wysiłek programisty przeniesiono na komputer wysyłający, na którym prawdopodobnie powinno znajdować się więcej programów. Program COMHEX został napisany w TURBOPASCAL'u i jego celem jest generacja zbiorów w formacie HEX firmy INTEL. Format ten zwykle produkowany podczas as- semblacji programów jest formatem pośrednim. Dopiero zastosowanie komendy LOAS, COM powoduje uzyskanie zbioru wynikowego typu COM, który może być uruchomiony. Wygodniej jest dostosować się do tego formatu niż pisać program tłumaczony na komputerze docelowym.

W programie zastosowano format rekordu typu absolutnego, który posiada następującą strukturę:

:	— znak startowy
ll	— bajt długości rekordu (20 hex w programie)
aaaa	— dwa bajty adresu ładowania
00	— zerowy bajt oznaczający rekord typu absolutnego
11 razy xx	— 11 bajtów w rekordzie
cs	— bajt sumy kontrolnej
<lf><cr>	— znaki końca linii i przejścia do nowej linii

np.: 08 01 00 00 xx xx xx xx xx xx xx xx cs<lf><cr>
(odstępy między parami są tylko dla jasności przykładu)

Ostatni rekord zbioru HEX musi składać się z 10 zer poprzedzonych znakiem „:"

W eksperymentach tego typu przeniesiono TURBO-PASCAL razem ze zbiorami dodatkowymi i instalacyjnymi z komputera Amstrad PCW 8256, wyposażonego w interface CPS 8256, na komputer SuperBrain QD II, pracujący pod systemem CP/M 2,2.

Stosowano kabel wg następującego schematu:

1	1	protective ground	(masa dodatkowa)
7	7	signal ground	(masa sygnałowa)
2	2	transmit data	(wyślij dane)
3	3	receive data	(odbierz dane)
4	4	requestto send	(żądanie wysłania)
6	6	data set ready	(dane gotowe)
8	8	data carrier detect	(detekcja nośnika danych)
5	5	clear to send	(wyzeruj, aby wysłać)
20	20	data terminal ready	(terminal gotowy)

przebieg testowy

```
A>comhex
zbiór do przekształcenia: c
Record(y): 1 Byte(y): 128
ilość generowanych zbiorów: 1
nazwa generowanych zbiorów: chex
128 bytes Record no. 1
:200100005E27455343275E273135271A1A1A1A1A1A1A1A1A1A1A1 A24
:200120001A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1 A7F
:200140001A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1 A5F
:200160001A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1A1 A3F
:0000000000 zbiór c (podlegający przekształceniu)
"ESC"15
```

Jonasz Mayer

```
program ComHex;
(*********************************************************)
(* program do konwersji pliku typu .COM na pliki typu .HEX *)
(*          zgodne z formatem firmy INTEL                *)
(*               (C) J. M.                               *)
(*              grudzien  1987                           *)
(*********************************************************)
const
    Bufsize = 64;     Recsize = 128;
type
    HexSTR = string(,4,);
var
    Source                ; file;
    Dest                  ; text;
    Buffer                ; array(,1,,BufSize,1,,,Recsize,) of byte;
    RecsRead, RecsToProc,
    FileNo, address       ; integer;
    SourceNane, DestName,
    CurrDestName          ; String(,20,);

function Hex(Number : integer; Bytes : integer) : Hex5ir;
const
     T   : array(,0,,15,) of char = '0123456789ABCDEF';
var
     D   : integer;      H   : HexStr;
begin
      H(,0,) := chr(Bytes+Bytes);
      for d := Bytes + Bytes downto 1
      do begin
           H(,d,) ; = Ti.Nuaber and 15,!;
            Number := Number shr 4;
         end;
      Hex := H;
end; (* of Hex *)

procedure ProcessBuffer;
var ByteToProc : byte;
    r,sr,i, chsum : integer;
begin
     for r := 1 to RecsRead
     do begin
     writeln('128 bytes Record no. ',r);
     for sr := 0 to 3
     do begin
                    write(dest,' :20');  write(';20');
                    write(dest,Hex(address, 2),'00');
           write(Hex(address,2),'00');
                    chsum := hi(address) + lo(address)+32,
                    for i := 1 to 32
                    do begin
                           ByteToProc := Buffert(,r,32*sr+i,);
                           chsum := chsum+ByteToProc;
                           write(dest,Hex(ByteToProc,1));
write(     Hex(ByteToProc,1));
        end;                              (* of Hex Rec Loop *)
      chsum := ((chsum mod 256)+1)*256 - chsum;
      writeln(dest,Hex(chsum,1));
      writeln(     Hex(chsum,1));
address := address + 32;
     end;                                 (* of SubRecs Loop *)
end;                                      (* of Recs Loop *)
if RecsToProc=0
then begin
          writeln(dest,':0000000000'),
          writeln(     ':0000000000');
      end;
end; (* of ProcessBuffer *)

begin (* main *)
write(' zbior do przekszta1cenia: '); readln(SourceName);
assign(Source,SourceName);
reset(Source);
RecsToProc := FileSize(Source);
writeln('Record(y); ',RecsToProc,'  Byte('',y): ',128,0*RecsToProc;4;0);
writeln('Liczba generowanych zbiorow; ',RecsToProc div BufSize+1);

write('Nazwa generowanych zbiorow: ');
readln(DestName);
reset(Source);
FileNo := 0;
address := $100;

While RecsToProc>0
do begin
     BlockRead(Source,Buffer,BufSize,RecsRead);
     FileNo := FileNo + 1;
     CurrDestName := Destna,e + ',00' + chr(ord('0')+FileNo);
     RecsToProc := RecsToProc - RecsRead;
     assign(Dest,CurrDestName);
     rewrite(Dest);
     ProcessBuffer;
     close(Dest);
   end;
  close(source);
end.
```

CHIMERA

Wokół Ziemi orbituje dziwny statek kosmiczny. Jest pozbawiony załogi i wyposażony w dziwne urządzenia. Twoim zadaniem jest, po dostaniu się na pokład statku, znalezienie czterech głowic, które pomogą w odczycie zakodowanej wiadomości dla mieszkańców Ziemi. Odczytasz ją w czterech niebieskich pokojach (BLUE ROOM) w kwadratach F5, G6, B6 i B8.

Grę rozpoczynamy w pomieszczeniu B3. Idziemy do pokoju E3, gdzie ukryty jest klucz do usuwania zapór energetycznych, i teraz dopiero rozpoczynamy właściwą grę.

Potrzebne głowice znajdziemy w pomieszczeniach C1, G7, C6 i G3.

Po zlikwidowaniu zapór energetycznych bierzemy pierwszą głowicę z pokoju C1 i zanosimy ją do niebieskiego pokoju F5. Pamiętajmy — na wskaźniku po lewej stronie u dołu ekranu widać symbol przedmiotu, który aktualnie niesiemy. Podnosimy je i zostawiamy lub używamy przez wciśnięcie przycisku FIRE. Chleb (BREAD) i wodę (MUG) przenosimy tylko wtedy, gdy nie niesiemy innych przedmiotów. W innym przypadku nastąpi natychmiastowe spożycie powodujące zwiększenie siły lub ochłodzenie robota.

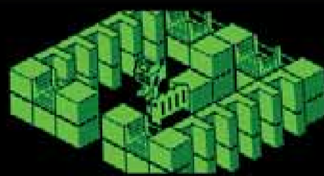
Zostawiając głowice w niebieskich pokojach pamiętaj aby nie zbliżyć się zbyt blisko chleba, który znajduje się w tym pomieszczeniu. Po zaniesieniu głowicy z C1 do F5 należy szukać następnej głowicy. Przeszkadza nam w tym toster blokujący przejście (G4, G8). Najpierw usuwamy toster z kwadratu G8. Pojawi się woda (na razie nie pij jej — usuń na- następny toster). Za tosterem w kwadracie G4 jest ukryty klucz (KEY), który otwiera drzwi (DOOR) w kwadratach D1, F8, H3 i A5. Klucz otwiera również pudełka (BOX) w kwadratach E7, E6, D7 i A7. Po otwarciu drzwi i pudełek możemy wziąć następną głowicę z G7 i zanieść ją do któregoś z wolnych niebieskich pokojów. Możemy też wziąć latarkę (TORCH) z pokoju A6. Idziemy z nią do ciemnego pokoju G3. Po zapaleniu latarki wejdziemy w posiadanie kolejnej głowicy, którą zanosimy do wolnego niebieskiego pokoju. Tu mała uwaga. W DARK ROOM możemy wypić wodę tylko przed zapaleniem światła. W innych przypadkach woda zniknie, musimy więc zrobić to po omacku.

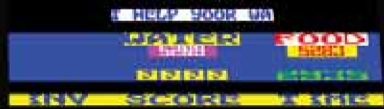
Wyruszamy teraz do kwadratu G2, gdzie ukryta jest piramidka służąca do usuwania klepsydr. Klepsydry te znajdują się w pomieszczeniach F3, E4, H7 i C7. Po usunięciu klepsydry w kwadracie C7 dostaniemy się do pomieszczenia C6, gdzie ukryta jest ostatnia głowica. Zanosimy ją do ostatniego wolnego niebieskiego pokoju. Pozostaje nam jeszcze przejść do zielonego pokoju (GREEN ROOM), który znajduje się w kwadracie A4.

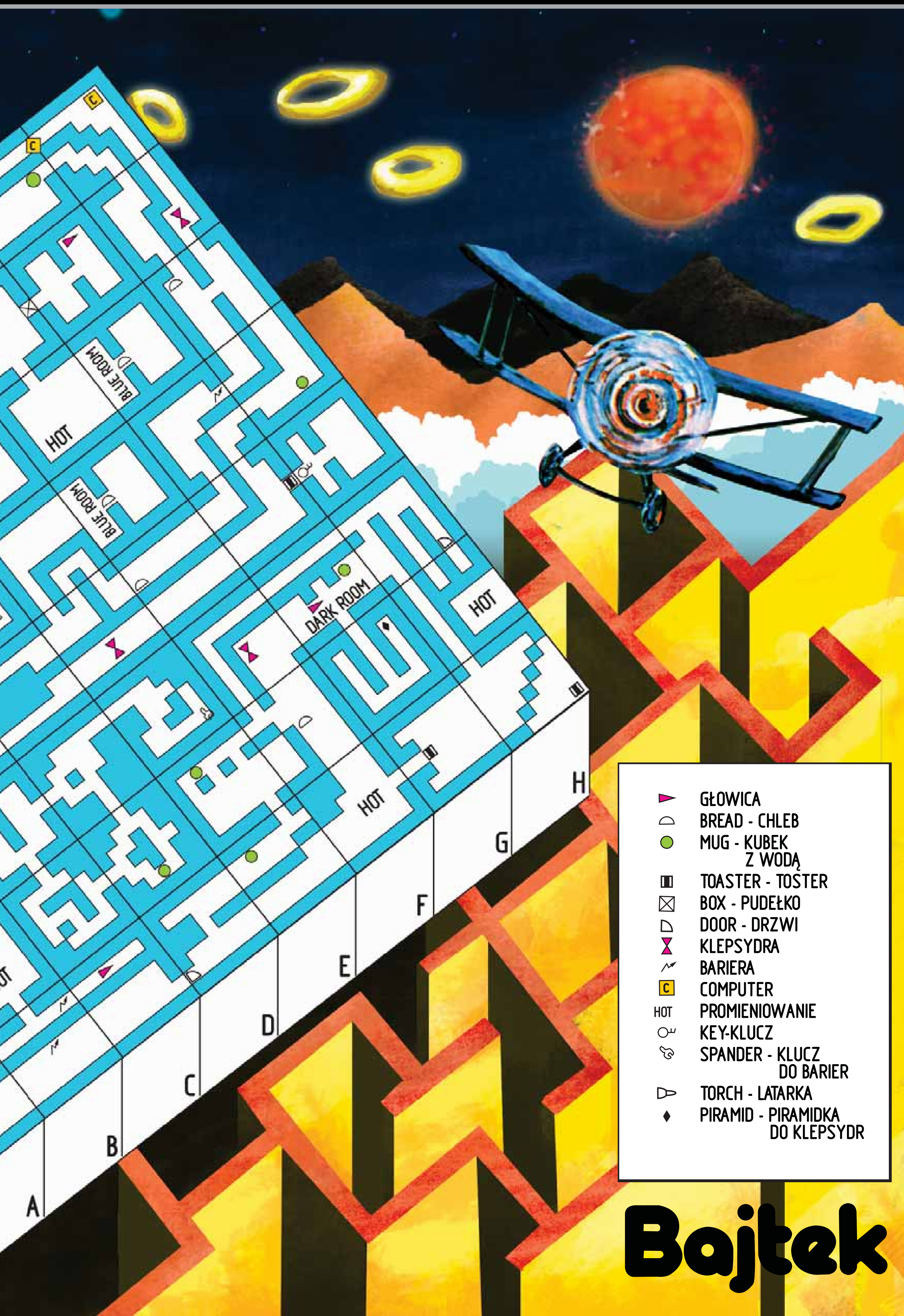
W nagrodę otrzymujemy nową grę oraz sposób jej uruchomienia (w moim przypadku nie chciała się uruchomić — zobaczyłem tylko czołówkę).

Nie podaję kolejności poruszania się po komnatach — pomyślmy nad tym pamiętając, że powinniśmy chodzić najkrótszymi trasami, w odpowiednim momencie zbierać chleb i wodę oraz unikać: pomieszczeń z promiennikami ciepła (HOT) — niestety, musimy wielokrotnie przechodzić obok nich. Nie wszystkie zapory, tostery, pudełka i klepsydry musimy otwierać lub niszczyć. Popatrz na plan i opracuj najbardziej ekonomiczną trasę.

Marek Filipczak






BLUE ROOM
HOT
BLUE ROOM
DARK ROOM
HOT
HOT
A
B
C
D
E
F
G
H
GŁOWICA
BREAD - CHLEB
MUG - KUBEK Z WODĄ
TOASTER - TOSTER
BOX - PUDEŁKO
DOOR - DRZWI
KLEPSYDRA
BARIERA
COMPUTER
HOT PROMIENIOWANIE
KEY-KLUCZ
SPANDER - KLUCZ DO BARIER
TORCH - LATARKA
PIRAMID - PIRAMIDKA DO KLEPSYDR
Bajtek

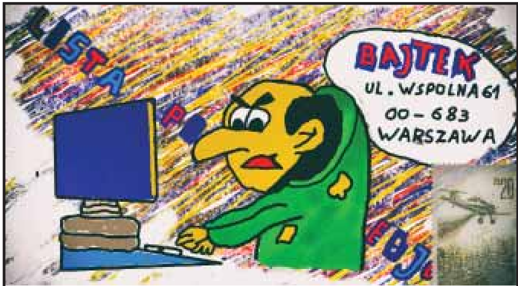


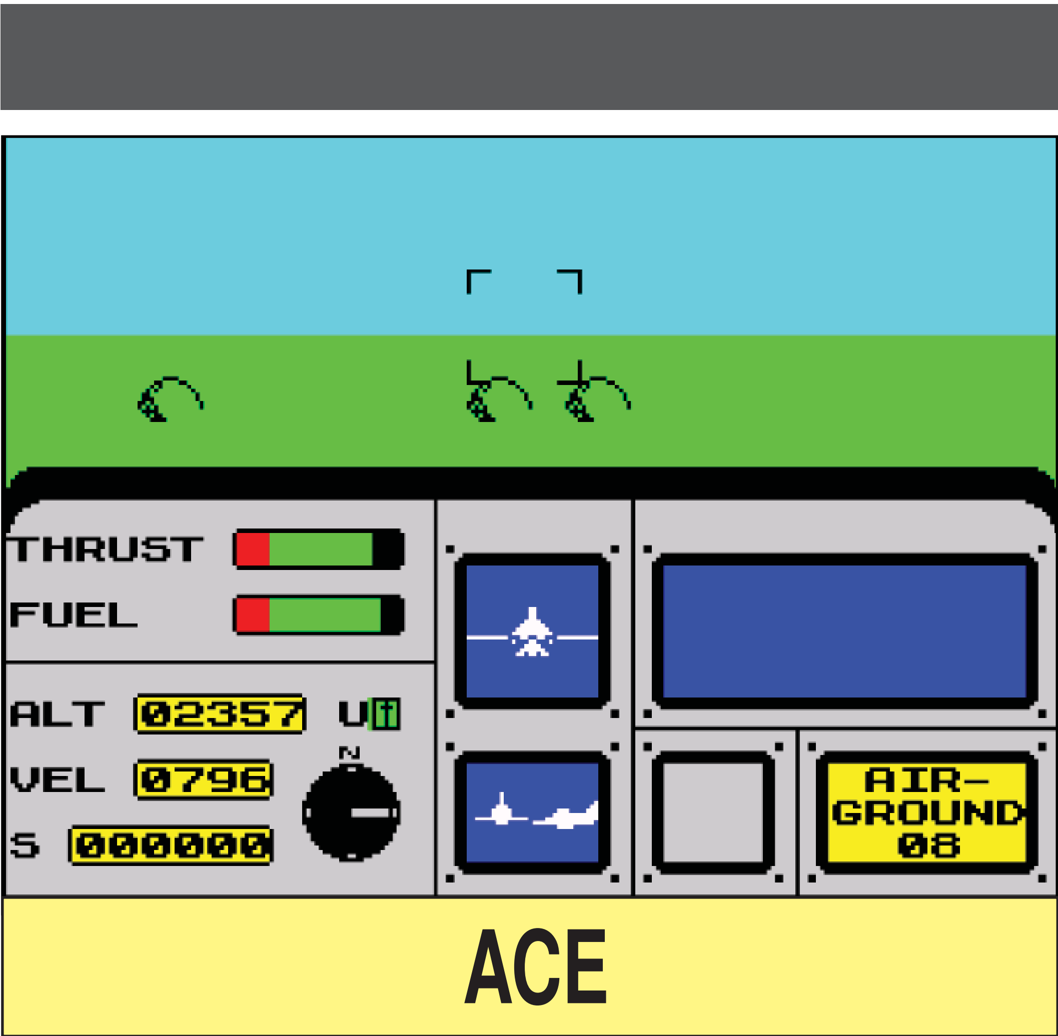
Nasza lista wciąż bez rewelacji. Pojawia się przecież tyle nowych, ciekawych gier. Widać jednak, że stare wciąż pozostają interesujące. Wciąż otrzymujemy nowe opisy — dziękujemy! Na obecne notowanie nadeszło 2971 propozycji na 152 tytuły.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	ARKANOID	!		x	x	x
2	SOLO FLIGHT		x		x	
3	TURBO ESPRIT	↑		x	x	x
4	SABOTEUR II	↓		x	x	x
5	GHOST CHASER	↓	x		x	x
6	STARQUAKE	↑	x	x	x	x
7	CHIMERA	↓	x	x	x	x
8	EXOLON	↓		x		x
9	GUN FRIGHT	↓		x	x	x
10	SHOW JUMPING	!		x	x	x

Nagrody otrzymują:
Anna Bielik i Paweł Walczak

W tym miesiącu kopertę nadesłał Aganowicz Grzegorz z Gliwic.





ACE

Spora część gier to gry symulacyjno-strategiczne. Większość tychże,to symulacje lotu walki powietrznej. Gra nazwana ACE jest symulatorem odrzutowego myśliwca Grumman F-14 Tomcat. Jest to jeden z najlepszych i najnowocześniejszych samolotów bojowych. Symulujący go program doskonale odwzorowuje reakcje maszyny w najróżniejszych sytuacjach — nurkowanie, pętle, beczki, walka powietrzna.

Na początku gry istnieje możliwość wybrania pory roku, poziomu trudności oraz załogi samolotu — sam pilot lub pilot i strzelec pokładowy. Teraz wybieramy jedno z czterech zadań: atak celów naziemnych, celów nawodnych, sił powietrznych lub wszystko naraz. W zależności od wykonywanego zadania otrzymujemy odpowiedni rodzaj broni. Mogą to być: działko 20 mm, pociski powietrze-powietrze, powietrze-ziemia, powietrze-woda oraz flary.

Po wytoczeniu maszyny na pas startowy należy natychmiast zwiększyć moc silników do ok. 3/4 pełnej mocy, przy prędkości ok. 250 mil/h poderwać się w powietrze i natychmiast schować podwozie. Rozbieg nad pasem startowym trwa mniej więcej 30 sekund. W tym czasie samolot kontroluje wieża i pilot nie ma możliwości sterowania samolotem oprócz zmiany mocy silników i wysokości lotu.

Pulpit sterowniczy podzielony jest na sześć części. Pierwsza od lewej to część informująca o ilości paliwa, mocy silników i rezultacie lotu. Dwa okienka w środku pokazują położenie samolotu w stosunku do poziomu i pionu oraz otoczenie maszyny (wrogie siły powietrzne i cele naziemne). Pozostałe okna to komputer pokładowy informujący o ewentualnym niebezpieczeństwie, obecności samolotu-tankowca i innych warunkach lotu a także widok do tyłu oraz rodzaj używanej broni i ilość pozostałej amunicji.

W dowolnym momencie można spojrzeć na mapę terenu. Jest to część lądu i kawałek morza. Na lądzie znajdują się trzy sojusznicze bazy oznaczone kolorem białym. Znad morza nadlatują wrogie samoloty a lądem ciągną dywizje nieprzyjaciela. Zadaniem pilota jest powstrzymanie naporu wrogich wojsk w głąb lądu i zajęcie całego terenu nadmorskiego. Wróg będzie dążył do okrążenia baz i zdobycia lądu. Zdobyta przez niego część terenu ma kolor ciemny, Twoja część — jasny.

W ogniu walki pamiętaj o szybko kończącym się paliwie. Na szczęście (poinformuje Cię o tym komputer) Twoi sojusznicy wysłali na pomoc samolot-cysternę. Tankowania paliwa możesz dokonać podlatując do niego ostrożnie z tyłu. Cysterna leci na wysokości 20500 stóp z prędkością 232 mil/h. Gdy samolot Twój będzie się już palił, pamiętaj, że pozostaje Ci możliwość katapultowania się, lecz nie zawsze dasz radę dotrzeć do bazy.

Komputer: Commodore 64/128, Spectrum 48/+

(m.p.)



Anna Bielik, lat 12
Gimnazjum im. Helmholtza w Essen (RFN)
zainteresowania: książki, plastyka
posiadany komputer: Atari XE-Game
wymarzony komputer: Amiga 2000
ulubiona gra: Reversi



Paweł Walczak, lat 9
Szkoła Podstawowa nr 75 w Warszawie
zainteresowania: piłka nożna, książki, pisanie własnych programów
posiadany komputer: Atari 800XL
wymarzony komputer: IBM
ulubiona gra: Zorro, Silent Service

S.O.S.

Proszę o pomoc w grze TIRES na Atari — nie wiem, jak przejść trzecie jeziorko.

Paweł Premicz
ul. Matejki 43/8
66-400 Gorzów Wlkp.

Pomocy! Nie mam pojęcia, jak poszukiwać nieśmiertelności w grach na Atari 800 XL. Liczę na waszą pomoc!

Tomasz Waligóra
ul. Leszczyńska 12
64-140 Włoszakowice

Mam Atari 65 XE. Proszę o dokładny opis gry ZORRO. Czy możliwe jest uzyskanie nieśmiertelności w grach: BRUCE LEE, MONTEZUMA'S REVENGE, ZORRO, FORT APOCALIPSA.

Ireneusz Mianowski
ul. Płk. Dąbka 69/17
82-300 Elbląg

Poszukuję opisów i nieśmiertelności w grach na Atari.

Robert Szmit
ul. Fantastyczna 5/1
20-531 Lublin

Pragnąłbym bardzo uzyskać nieśmiertelność w grze CHUCKIE EGG w wersji na Spectrum lub Amstrada 464 i w grze BOMB JACK w wersji na Timex 2048. Poszukuję także gry KUNG-FU MASTER na Timexa lub Amstrada i jej mapy wraz z opisami (także ułatwień).

Jacek Cholewa
ul. 12-ego lutego 11/16
82-300 Elbląg

Jak zakończyć gry NINJA i SPY HUNTER w wersji na Atari? Szukam także opisu gier ZORRO i JUMBO JET na ten sam komputer.

Piotr Gniewkowski
ul. Wolskiego 41/89

Pilnie poszukuję opisów gier: ATIC ATACK, ZORRO i JAMES BOND na ZX Spectrum + .

Michał Laudański
ul. Dantego 7/39
01-914 Warszawa

Mam Atari 65 XE. Nie wiem, jak przejść trzeci etap w grze MINER 2049 i szósty w NEPTUN'S DAUGHTERS.

Krzysztof Przeorski
ul. Dywizjonu 303 33F/30
80-462 Gdańsk-Zaspa

Poszukuję dokładnego opisu gry UNIVERSAL HERO w wersji na Atari 800 XL. W zamian odstąpię opis gry RIVER RAID i HERO.

Piotr Myjak
ul. Dobrego Pasterza 125/42
31-416 Kraków

Bardzo proszę o pomoc w grach SIR FRED i RIDDLES DEN.

Jarosław Mucha
ul. Jasia i Małgosi 24
94-118 Łódź

Usilnie szukam opisu do gry STRIKE FORCE COBRA, a także nieśmiertelności do niej i do FAIRLIGHT i ANTIRIAD. W zamian oferuję Poke do gier: CAULDRON II, BEACH HEAD, BOMB JACK i innych na Commodore 64.

Wiktor Długosz
ul. Kosynierów 2g/4
41-706 Ruda Śl. 6

Jak uzyskać nieśmiertelność w grach: BOMB JACK, DYNAMITE DAN i COMMANDO na Spectrum 48. Szukam też loaderów do gier URIDIUM i FIRE LORD.

Adam Operhalski
ul. Piastów 24/189
40-868 Katowice

W jaki sposób uzyskać nieśmiertelność w grach: MONTEZUMA'S REVENGE, GREEN BERETS, JET SET WILLY i BLUE MAX w wersji na Atari 65 XE?

Paweł Walczak
ul. Królewska 2/30
00-065 Warszawa

YACHT RACE

Nazwa niedwuznacznie wskazuje, że jest to gra na długie zimowe wieczory. Zresztą nie tylko zimowe. W sierpniu, marząc o wyrwaniu się z domu na Mazury, w październiku, wspominając minione, a w marcu planując nadchodzące wakacje, włączcie komputer, by przybliżyć sobie wrażenie kołysania, powiewu wiatru, i uchwytu rumpla. Oczywiście prawdziwemu żeglarzowi nawet najlepsza gra nie zastąpi choćby „Omegi" na wodzie.

Yacht Race to regaty „finem" po małym jeziorze. Do dyspozycji mamy tylko ster i własny rozsądek: jak hałasować, by okrążyć boje w jak najkrótszym czasie. Uwaga na wraki, kamienie i mielizny!

Po ukończeniu rejsu możemy zwiększyć nieco stopień trudności — zaczynamy regulować ustawienie żagla. Teraz zabawa jest o wiele ciekawsza, gdyż wskaźniki po lewej stronie ekranu pokazują, jak możemy wykorzystać żagiel, a na ile go wykorzystujemy. Gdy i to okaże się zbyt nudne, proponuję dołączenie elementu balansowania na jachcie. Całość jest teraz obsługiwana przez klawisze (ZX Spectrum): 0,P — ster; C,V — zmiana kierunku widzenia; I,N — ustawienie żagla; Q,W — przechył jachtu; E — mapa.

Czyżby to było wszystko, co ma do zaproponowania ten program? Oczywiście nie! Podałem trzy możliwości żeglowania, a każdą z nich można teraz ciekawie urozmaicać. Prawdziwe regaty zaczynają się bowiem dopiero od poziomu czwartego (LEVEL 4), gdyż razem z nami startuje przeciwnik. Niestety przeciwnikiem kieruje wyłącznie komputer, ale to powinno być zachętą dla ambitnych. Po co udowadniać swą wyższość nad młodszą siostrą, spróbujcie pokonać Black Streak.

Gra na początku wyda Wam się z pewnością bardzo wolna i monotonna, ale nie oddalajcie się w czasie gry od komputera. To odbicie prawdziwego żeglarstwa — mała chwila nieuwagi i... ekran niedwuznacznie zapełni się barwą, jaką zwykle podziwiamy z wysokiego brzegu. Na monotonię na pewno nie będziecie narzekać, gdyż faktycznie najwyższym poziomem gry jest LEVEL 6, COURSE 9. Jak zapewne szybko się zorientujecie, przy pomocy klawiszy SPACE I ENTER możecie wybierać jedną z dziewięciu tras (o różnych wiatrach), a także ilość okrążeń jeziora. Nie przejmujcie się słabą widocznością. W większości gier jesteście przecież powiadamiani o sukcesach lub porażkach. W przypadku tych drugich warto dokładnie przetłumaczyć sobie podane informacje.

Po przepłynięciu wielu, nawet najtrudniejszych tras nie przystępujcie natychmiast do egzaminu na patent sternika. Radzę zacząć od zdobycia karty pływackiej i uczestniczenia w obozie żeglarskim, gdzie pod okiem instruktorów sami dojrzycie różnice między pseudosymulacyjną grą, a rzeczywistością.

Komputer: ZX Spectrum 48/+, Commodore 64, Amstrad/Schneider 6128,464

(m.p.)

Na specjalne życzenie Czytelników przedstawiamy dziś mapę miasta Rock City — rodzinnego miasta Freda Flinstone'a. Na pewno, pomoże ona w poruszaniu się po tym, dość skomplikowanym mieście. Pamiętajmy, że aby przejść na inną ulicę, należy trzymać klawisz DÓŁ lub GÓRA przy przechodzeniu z lewa na prawo lub odwrotnie. Pieniądze na opłacenie dinozaura znajdziesz w prawym dolnym rogu miasta. Wilma porusza się losowo i trudno ją znaleźć; stanie się to łatwiejsze po zbudowaniu całego domu.

(b)



CIEŃ CeBITu

Reguła, znana każdemu miłośnikowi komputeryzacji mówi, że czym mniejsza jest firma, tym dumniejszą nosi nazwę. Podobnie bywa i z targami. Poznański „Infosystem", chociaż w założeniu miał być imprezą informatyczną numer jeden, i to nie tylko dla Polski, przynajmniej jeszcze w 1988 roku, potwierdził wspomnianą opinię.



Oczywiście przesadą byłoby mówić, że potencjalni wystawcy zlekceważyli targi. Wielu z nich zameldowało się w dużej sile. W porównaniu chociażby do warszawskiej wystawy „Komputer 88" widać było znacznie większą aktywność dużych firm państwowych i producentów z krajów socjalistycznych. Zawiedli natomiast organizatorów wystawcy zachodni, z których tylko nieliczni przedstawiali jakieś naprawdę konkretne oferty.

Czy „Infosystem" oferta na nim zaprezentowana pozwala z nadzieją myśleć o dniu jutrzejszym? W dziedzinie hardwaru raczej nie. Wprawdzie „Elwro" pokazało nareszcie „Juniora" nie tylko na stoisku, ale również w akcji, podczas lekcji prowadzonych w pawilonie edukacji komputerowej, a także całkowicie „żółty" w środku model 801 AT lecz na tym oferta rodzima się kończyła, jeśli nie liczyć kilku drobnych producentów, takich jak choćby „Comtech". Spółdzielnia ta rozpoczyna właśnie produkcję mi nikomputera CTCS 16/24, jak twierdzą jej prezesi, konstrukcji własnej zgodnej z linią PDP 11. Ten i kilka innych minikomputerów było jedynymi rodzynkami w „IBM-owym ciasteczku". Monokultura mikrokomputerów opartych o linię Intela jest u nas niemal absolutna.

Co ciekawe, wielu bossów rodzimej komputeryzacji uważa tę monokulturę za szczególnie pożądaną i pożyteczną. Podczas swej konferencji prasowej szefowie „Elwro" przyznali, że wystąpili do ministra finansów o wprowadzenie cła na przywożone do Polski mikrokomputery niezgodne ze standardem IBM (!). **Ciekawe, czy autorzy tego pomysłu robią to dlatego, bo darzą „Big Blue" platonicznym uczuciem, czy po prostu z głupoty? Chociaż bowiem IBM-y zdominowały rzeczywiście rynek światowy nikt przecież nie wie, czy właśnie ta konstrukcja jest najbardziej przyszłościowa.** Dlatego chociażby Amiga, czy Atari ST cieszą się także powodzeniem. Nikt poza tym nie mógł nam wyjaśnić, co ma we wspomnianym wniosku znaczyć słowo „mikrokomputer"? Także domowy? **A może, zdaniem „Elwro", trzeba także wprowadzić cło na maszyny typu „Vax", czy „Sun", które tylko dzięki importowi prywatnemu prześlizgują się dziś do nas przez sito embarga, a bez których, szczególnie w dużych przedsiębiorstwach, maszyny typu PC są jedynie zabawkami.**

Rozpisałem się nieco o sprawie, która została jedynie zasygnalizowana podczas targów, ale to chyba dlatego, że uwidoczniły one kilka niezbyt pocieszających tendencji. Jedną z nich jest wspomniana monokultura sprzętowa. Komputery tylko PC, drukarki tylko Star, itp. Są to wprawdzie produkty bardzo nowoczesne, a przyjęcie standardu jest niezbędne dla komunikacji między systemami, ale zawsze trzeba chociażby wiedzieć, co też dzieje się na innych kierunkach rozwoju.

Łatwizna, także finansowa (mało transakcji za to z jak najlepszym przebiciem) sprawiły, ża jak już pisaliśmy we wstępniaku do poprzedniego „Bajt- ka", praktycznie nieobecna była na targach oferta dla indywidualnego nabywcy. Wprawdzie mają się ponoć pojawić w sklepach „Juniory", ale cena 300 tys. lub więcej, za samą jednostkę centralną plus 700 tys. za stację dyskietek (ciekawe dlaczego ma ona rozmiar nesesera?) nie wydaje się czynić go „komputerem dla mas". Dlatego też z entuzjazmem niemal przyjęliśmy wiadomość o podpisaniu w Warszawie właśnie podczas trwania poznańskiej wystawy, kontraktu między „Pewexem", a dealerem koncernu Commodore. **Wprawdzie wciąż jeszcze za bony, ale trafi do nas w dużych ilościach kolejny po Atari prawdziwy mikrokomputer domowy i to nawet z wersją de luxe, czyli Amigą 500 dla bardziej zasobnych.** Commodore

◄ Elwro 801 AT- tak właśnie prezentuje się wrocławski AT z tajwańskim wnętrzem.

Systemy CAM prezentowane były na gustownych modelach. ▼



swoją polską premierę, będzie miał już w czerwcu, podczas Międzynarodowych Targów Poznańskich. Na „Infosystemie” go nie było.

Wspomniałem już o dość poważnym potraktowaniu wystawy przez duże firmy ze wszystkich krajów socjalistycznych. Niewątpliwie najpoważniejszą spośród nich ofertą pochwalić się może NRD-owski „Robotron”. W jej skład wchodzą m.in. dwa mikrokomputery klasy PC- A 7150 i EC 1834 minikomputery, drukarki i elektroniczne maszyny do pisania. Najważniejsze jest jednak, że wszystkie te maszyny produkuje się w dużych seriach i przy sporym udziale rodzimych podzespołów. Oczywiście wiele jest w nich także podzespołów zachodnich i dalekowschodnich, ale jest to już produkcja, nie zaś tylko składanie.

Trudno natomiast mówić o wielkoseryjnej produkcji tego, co pokazał na Infosystemie „Polkolor”. Trzy prezentowane tam typy kolorowych telewizorów a także magnetowid z nalepką „Polkolor- -Schneider” prezentowały się wprawdzie znakomicie, ale cóż z tego, skoro w sklepach nikt ich raczej nie oglądał i nieprędko pewnie zobaczy. Trzeba jednak przyznać, że wyglądały naprawdę nowocześnie a wspomniany magnetowid na głowę bije konkurenta z „Kasprzaka”. Czy jednak rodzimej produkcji nie jest w nim jedynie naklejka — trudno powiedzieć.

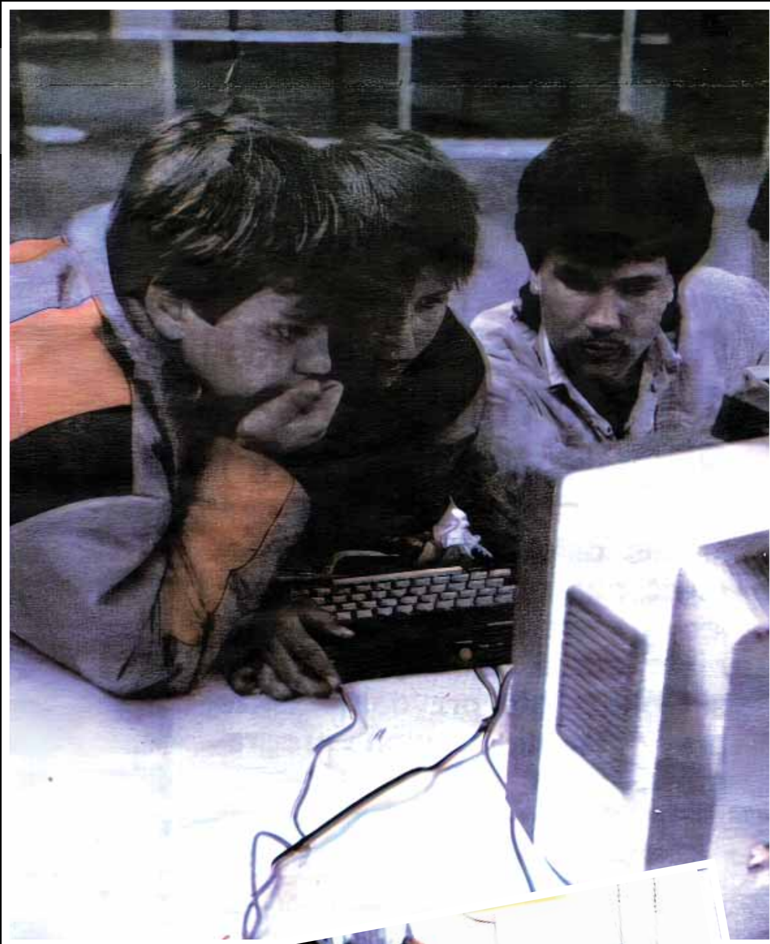
Niewielka aktywność na „Infosystemie” firm zachodnich nie oznacza, że na ich stoiskach nie było co oglądać. U „Philipsa” dla przykładu szokował zupełnie nowy system składający się z mikrokomputera wyposażonego m.in. w stację CD-ROM (płyt kompaktowych) i gramowidu. Pozwala on na rejestrację na dwóch nośnikach — małej i dużej płytach optycznych programu edukacyjnego zawierającego dla przykładu całą encyklopedię z ilustracjami.

Pojawiły się także na naszym rynku firmy wprawdzie bardzo znane, lecz dotąd jakby omijające Polskę. Zdecydowaną supremację „Stara” w dziedzinie drukarek podważyć chcą m.in. Epson i Citizen. Pierwszy szykuje ponoć ofensywę w swoim, czyli wielkim stylu, drugi zdobywa teren powoli za sprawą zachodnioniemieckiego pośrednika- firmy Synelec. M.in. jego drukarki zobaczyliśmy na stoisku spółki „Mikrokomputery”. Dziwi tylko, że za profesjonalne uważa się u nas wyłącznie drukarki o szerokości wałka 15 cali. Tymczasem obok MSP 15E „Mikrokomputery” mogłyby dla swoich klientów zakupić, także chociażby testowane przez nas Citizeny 120D. **Jakoś w przypadku maszyn do pisania format A4 nam odpowiada. Dlaczego po zainstalowaniu komputerów potrzebne są od razu większe płachty papieru?**

Na koniec o tym, co powinno stanowić chyba podstawę oferty targowej — o softwarze. Nie było tutaj także żadnych rewelacji. **Swoją drogą ktoś mógłby, poza nami, zastanowić się dlaczego tak się dzieje? Do dziś niewiele powstało u nas choćby tak podstawowych programów jak edytor tekstu (na „Infosystemie” były prezentowane na poważnie tylko dwa), nie ma żadnego systemu Desktop Publishing, brakuje najprostszych rodzimych programów narzędziwych.** Trochę lepiej jest natomiast w dziedzinie specjalistycznego oprogramowania użytkowego, pojawiły się pierwsze systemy CAD CAM.

Potrzeba nam z pewnością więcej takich imprez jak „Infosystem”, czy „Komputer”. Przegląd własnych możliwości w cieniu CeBITu, czy choćby odbywającego się równolegle z imprezą poznańską paryskiego SICOBu pozwala nam, spróbować odpowiedzieć na pytanie, jak też ma się nasza komputeryzacja? Moja prywatna najkrótsza odpowiedź byłaby chyba jednak dość enigmatyczna: trochę dobrze, trochę źle.

Grzegorz Onichimowski



Z największym napięciem wpatrywali się w ekrany monitorów najmłodsi z widzów ►



W pawilonie edukacji komputerowej prowadzono normalne zajęcia dla kilku poznańskich klas. Można było zobaczyć w akcji pełne zestawy "Juniorów". ►



▲ Stoisko IBM nie było zbyt bogate. Z rodziny PS2 prezentowane były tylko widoczny na zdjęciu model 50 oraz model 30

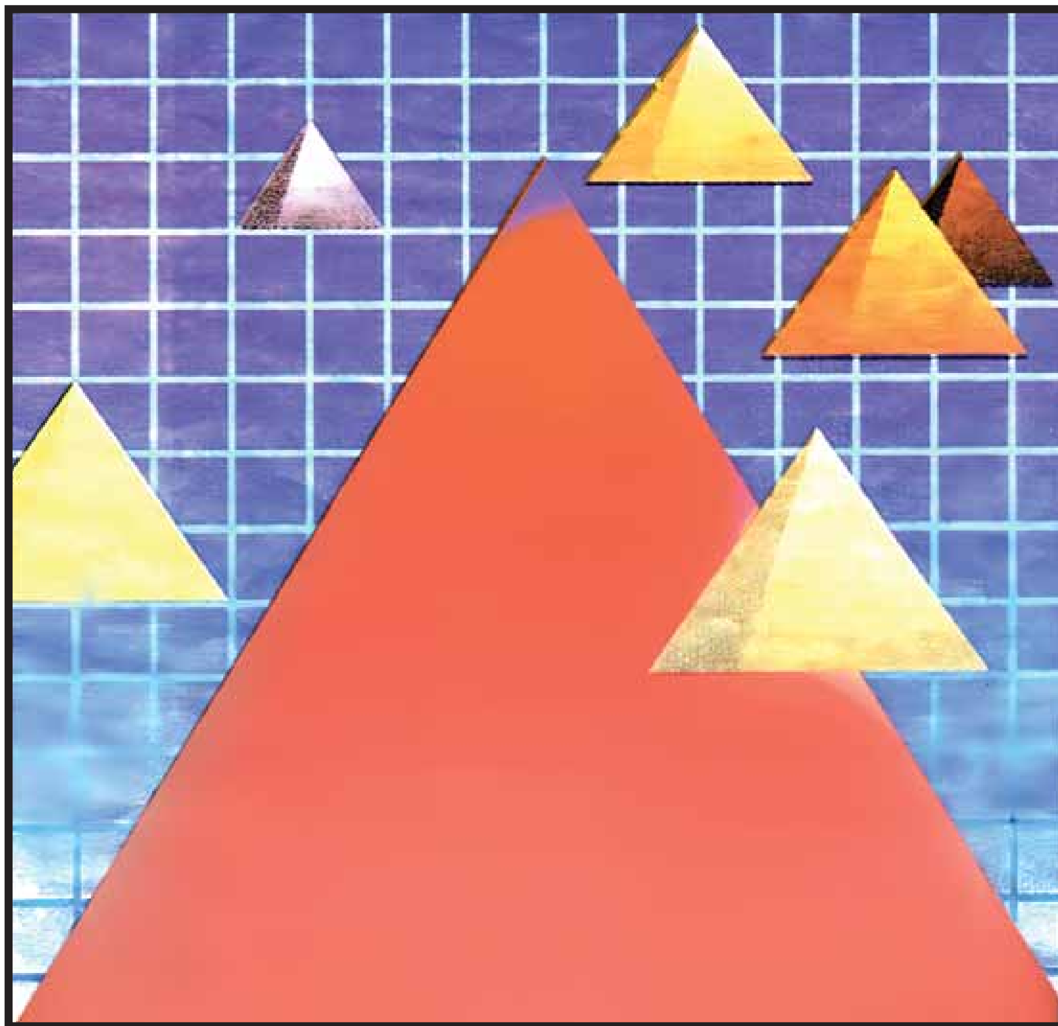


W trakcie trwania wystawy programiści demonstracyjnie tkwili na posterunkach prac ►

ZUPA Z GWOŹDZIA

(TYM RAZEM O SMAKU WYBITNIE PROGRAMISTYCZNYM)

Dziś dalej będziemy się pastwić nad programem obliczającym wartość wielomianu, mam jednak nadzieję, że wnioski okażą się przydatne także przy pisaniu innych programów.



Przypomnijmy sobie na początek część programu, w której rozprawialiśmy się z problemem dynamicznego wykorzystania początkowego fragmentu tablicy:

```
program hor (input, output);
const nmax = 20;
A:array [0..nmax] of real;
readln (n); if n>nmax then
writeln ('Blad — max. dopuszczalna wartość n wynosi'i, nmax)
```

Zwróćmy uwagę, że górna granica dopuszczalnych wartości n została ustalona na 20. Mogliśmy więc napisać po prostu:

A: array [0..20] of real; i dalej odpowiednio:

if n>20 then

Jednak użycie stałej nmax = 20 jest korzystniejsze, gdyż jeśli zajdzie potrzeba zmiany wartości 20 np. na 30, to dokonamy tej zmiany w jednym miejscu (nma- x = 30;) a o resztę programu nie musimy się martwić.

Przejdźmy teraz do uruchamiania i testowania programów. Zwykle jedno i drugie wymaga wykonania programu wiele razy, zanim nie wykryjemy i nie usuniemy wszystkich błędów. Każde wykonanie programu wymaga wczytania danych. Jeśli dane składają się z kilkudziesięciu liczb*), to ich wpisanie przez człowieka zajmuje sporo czasu. Chyba że program sam stworzy sobie dane do testowania. Właśnie taką metodę zastosowaliśmy w naszym programie. Pętla:

for i = 0 to n do A [i] : = i; zaoszczędza nam wpisywania współczynników wielomianu we wczesnej fazie testowania. Inne możliwości wypełnienia tablicy danymi do testów to np.

for i = 1 to n do A [i] :=1; (lub zamiast 1 inna stała wartość)

for i = 1 to n do A fi] : = 0; (zwykle dla takich danych programy zachowują się w szczególny, często ciekawy sposób, np. wielomian powinien mieć wartość 0 niezależnie od wartości x).

Można też do elementów tablicy wpisywać liczby pseudolosowe. Większość języków programowania stawia do dyspozycji użytkownika funkcje, która po wywołaniu zwraca wartości wylosowane z pewnego przedziału liczb rzeczywistych.**)

Oczywiście możliwości jest tu praktycznie nieograniczona ilość, przyjęte rozwiązanie zależy zwykle od specyfiki zadania i od inwencji programisty. Pamiętajmy tylko o dwóch ważnych sprawach:

Dane testowe, wygenerowane przez program warto wyświetlić lub wydrukować razem z wynikami. Inaczej nie będziemy mieli pewności czy błędny wynik powstał w wyniku błędu w obliczeniach, czy też pomyliliśmy się we fragmencie generującym dane.

Ostateczne testowanie programu powinno być prowadzone dla danych wczytywanych i to przynajmniej z dwóch powodów. Po pierwsze, dane utworzone automatycznie mają zwykle regularną budowę, są „sztuczne". Warto zrobić parę testów na danych prawdziwych, wziętych z dziedziny, w której program będzie stosowany. Po drugie, trzeba przecież przetestować fragment wczytujący dane!

Teraz zmieniamy temat, choć zostajemy przy programowaniu. Wyobraźmy sobie, że przyszedł do mnie kolega i rzecze tak: „Słyszałem, że masz fajny program na obliczenie wartości wielomianu. Coś takiego jest mi potrzebne do mojego systemu projektowania samochodów. Dasz mi?" Bardzo chętnie, chyba każdy autor chce rozpowszechniać swoje programy. Tylko jak to zrobić? Okazuje się bowiem, że kolega ma swój system już prawie gotowy, czytanie danych i wydruk wyników nie są mu potrzebne, za to zmiennych o nazwach i, n, w, x, A używa do zupełnie innych celów. W tej sytuacji zwykłe dopisywanie tekstu naszego programu do jego części jest wykluczone, trzeba by dokonać sporych przeróbek. Zamiast tego kolega woli napisać sobie potrzebny kawałek od początku. On straci trochę czasu, a ja okazję zabłysnąć swoim dorobkiem.

Co w takim razie miałby zrobić np. pan Horner, gdyby żyjąc w naszych czasach wymyślił świetny algorytm i chciał go opublikować w formie dostępnego dla wszystkich programu? Oczywiście zapisać swój algorytm w postaci procedury. Stopień i współczynniki wielomianu, wartość x, oraz wynik w zostaną przekazane jako parametry. „Popatrzmy na program program

```
main (input, output);
{program główny do testowania procedury Homer}
const
   nmax = 120;
type
   tabl = array [0 ..nmax] of real;
var
   x, w :real;
   n, i :integer;
   WIEL : tabl;
procedurę Horner (A: tabl; n: integer; varx, r: real); {obliczenie wartości wielomianu stopnia n, zapisanego w tablicy A zwraca r — wartość w punkcie x, obliczoną algorytmem Homera}

var
   i :integer;
begin
   r:=A[n],
   for i: = n — 1 downto 0 do
   r: = r* x + A [i];
end;   {procedury Horner}
begin {początek programu}
{przygotowanie danych wygląda tak jak poprzednio}
writeln ('podaj stopień n');
readln (n);
if n>nmac then
   writeln ('Błąd — max. dopuszczalna wartość n
   wynosi'', nmax) else
    begin
   {generacja testowych współczynników wielomianu}
    fori: = 0 ton do WIEL [i] :=i;
    writeln (Podaj x');
    readln (x);
    {dane przygotowane, wywołujemy procedura}
    horner (WIEL, n,x,w);
    writeln ('Dla x= ',x,' wartość wielomianu wynosi', w)
    end
end.
```

Obliczenie wartości wielomianu znalazło się w treści procedury Horner. Procedurę tą możemy wykorzystać wielokrotnie w tym samym programie — za każdym ra zem będzie to tylko jednak instrukcja wywołania, więc jeśli potrzebujemy skorzystać z algorytmu Homera w kilku miejscach, to nasz program stanie się dzięki

użyciu procedury krótszy i czytelniejszy. Jest to niewątpliwy zysk, o którym zawsze trzeba pamiętać programując, ale w opisanej powyżej sytuacji ważniejsze jest co innego. Procedurę Homer można przenieść zupełnie bezboleśnie do innego porogramu ***). Nie wystąpią żadne kolizje nazw (zmienna pomocnicza „i" wewnątrz procedury, to zupełnie inna zmienna niż „i" w programie głównym).

Tak więc dzięki procedurom możemy w wygodny sposób tworzyć biblioteki typowych programów, mogące ułatwiać później pracę wielu programistów.

Zauważmy także, że w treści procedury nie znalazły się: wczytywanie danych i wydruk wyników. Nie są tam potrzebne, tylko zaciemniały przedstawienie algorytmu. Rzeczy tego typu nawet początkujący programista potrafi zaprogramować, dlatego proponuję następującą umowę: jeśli będę w przyszłości opisywał jakieś programy, to wczytywanie danych, wydruki wyników itd. będę pomijał. Każdy pamięta, że są niezbędne, każdy wie jak to robić, więc szkoda na to miejsca.

Przy okazji, skoro przerobiliśmy dobrze działający program na nową wersję, to trzeba tę nową wersję także przetestować. Jeśli poprzedni program był dokładnie przetestowany, to teraz możemy sobie nieco uprościć życie i zamiast sprawdzać wyniki ręcznie, puścić oba programy dla kilku zestawów takich samych danych i porównać wyniki.

Skoro już porównujemy wyniki dwóch programów rozwiązujących to samo zadanie, to przypomnijmy i wykonajmy dla tychże danych program obliczający wartość wielomianu sposobem tradycyjnym. Porównujemy otrzymane wyniki, dla części danych zgadzają się dokładnie, np.:

dla m=5; x=5; wynik obu programów wynosi 18555; Gdy weźniemy jednak np.

n = 100, x=1.1 otrzymujemy coś bardzo dziwnego: wyniki różnią się nieco:

7.2527869011 105 z algorytmu Homera, i

7.2527869012 105 z algorytmu zwykłego.

Co to może być, który wynik jest dobry? Spróbujmy wyciągnąć nasz stary program obliczający wartość wielomianu, napisany w BASIC-u. Dla tych samych danych otrzymujemy wynik:

7.252798 105, czyli jeszcze coś innego.

Spróbujmy w takim razie zebrać więcej materiału doświadczalnego do przemyśleń. Puszczamy nasz program pascalowy dla n=100, x=5 i otrzymujemy ciekawy komunikat:

Run-time error 01. PC=2DEF

Program aborted.

Komunikat ten oznacza niewątpliwie, że podczas wykonywania naszego programu nastąpił błąd, wykonanie zostało przerwane. Czy komputer lub program zostały uszkodzone? Weźmy jeszcze raz n=5, x=5. Wynik jest poprawny!

No cóż, ostatnie łyżki naszej zupy z gwoździa nabrały dość pikantnego smaku. Nas to jednak nie zraża. Zgodnie z tym co postanowiliśmy miesiąc temu nawet nie próbujemy podejrzewać tu komputera, ale od razu zaczynamy szukać błędu w programie. Tylko, że tym razem naprawdę nie widać żadnego błędu! Rzeczywiście, macie rację. Błędu w programie nie ma. Ale komputer też jest w porządku. To po prostu nasza wiedza o komputerze jest jeszcze niepełna. W tej sytuacji z wyjaśnieniem tych dziwnych, i co tu ukrywać, bardzo niemiłych fenomenów musimy poczekać do następnego miesiąca. Może zresztą, zanim miesiąc upłynie, sami wymyślicie wyjaśnienie?

---

*) w zadaniach praktycznych może to być więcej, np. kilka tysięcy liczb.

**) Najczęściej funkcja taka nazywa się RAND, RANDOM, lub podobnie.

***) Jedynym, zresztą chyba oczywistym, warunkiem jest aby nazwa Horner nie była w tym programie użyta do innych celów.

Andrzej Pilaszek

GIEŁDA (ceny na dzień 1988.04.24)

	GIEŁDA BAJTKA (tys. zl)	PEWEX BALTONA (USD)	Wielka Brytania (funty)
SINCLAIR			
ZX Spectrum	135	115	50
ZX Spectrum Plus	160	–	60
ZX Spectrum 128 kB + 2	280	–	100
ZX Spectrum 128 kB + 3	350	–	175
Drukarka SEIKOSHA GP 50S	110	–	40
TIMEX 2048	160	146	–
Joystick	6–10	4–12	3–12
COMMODORE			
C-16	90	–	55
C-116	75	–	45
C + 4	150	–	70
C64	235	219	155
C128	360	299	225
C128D	840	–	310
Amiga 500	1000	–	445
Magnetofon 1530, 1531	50	48	25
Stacja dyskietek 1541	275	–	170
ATARI			
65 XE	175	125	65
130 XE	275	199	100
XC 12	50	48	20
Stacja dyskietek 1050	220	185	99
LDW Super 2000	300	199	–
AMSTRAD			
464 z mon. monochromat.	320	–	155
464 z mon. kolor.	405	–	180
6128 z mon. mono.	590	–	230
6128 z mon. kolor.	685	–	310
Stacja dyskietek 3" do 464	300	–	115
PC 1512 SD MM	950	–	300

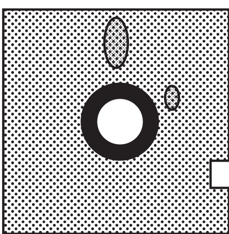
SPECTRUM W NATARCIU

Wśród posiadaczy domowych mikrokomputerów istnieje często specyficzna rywalizacja. Właściciel konkretnego mikrokomputera jest jednocześnie propagatorem swojego sprzętu, wychwalając jego zalety i ganiąc wady mikrokomputerów innych firm.

Największą popularnością wśród młodzieży nadal cieszy się Spectrum, ze względu na cenę i chyba najbogatsze oprogramowanie, zwłaszcza największy wybór gier. I właśnie dla zwolenników tego mikrokomputera adresowana jest informacja o pojawieniu się nowego modelu, na razie na rynkach zachodnich. Znając przedsiębiorczość prywatnych importerów niedługo pojawi się także na giełdzie „Bajtka". Nowość nazywa się Spectrum +3 i pod względem konstrukcji przypomina CPC 6128, zawiera 128 kB pamięci RAM i ma wbudowaną stację dysków 3 cale.

Lansowanie przez firmę Amstrad coraz to nowych wersji Spectrum przypominających Amstrady ma proste i zaskakujące wyjaśnienie. Otóż w kraju producenta czyli w Anglii większą popularność ma Spectrum niż Amstrad. co prawdopodobnie nie zmieni proporcji krajowych zwolenników, ale dostarczy kolejny argument do dyskusji.

(gr)



INDYWIDUALNY BANK DANYCH

Marta Jaśkiewicz, posiada komputer Philips P 2000 T//102. Prosi o pomoc w zdobyciu programów na wyżej wymieniony komputer. Adres: 61—674 Poznań, Oś. Kraju Rad 11 m 64.

Jacek Macierzyński, lat 19. Posiada Atari 800 XL oraz magnetofon XC-12. Oprogramowanie: programy użytkowe, narzędziowe i gry. Zainteresowania: saence-fiction, informatyka oraz muzyka. Proponuje wymianę doświadczeń i programów. Adres: 87—100 Toruń, ul. Bankowa 6/4.

Andrzej Zakutyński, posiada mikrokomputer C—16 oraz Atari 800 XL. Oprogramowanie: duży zbiór gier, programów użytkowych. Zainteresowany jest wymianą oprogramowania. Adres: 75—061 Koszalin, ul. Wróblewskiego 18/5.

Paweł Piec, lat 13. Posiada komputer YASCHICA YC-64 kB system MSX, magnetofon, joystick i monitor. Oprogramowanie gry i programy użytkowe oraz programy własne. Proponuje wymianę oprogramowania oraz doświadczeń. Adres: 44—117 Gliwice, ul. Gw. Polarnej 9/3.

Adam Adamczyk, lat 15. Posiada mikrokomputer Atari 800 XL, magnetofon XC. Oprogramowanie: około 80 gier oraz programy własne. Proponuje wymianę doświadczeń i programów. Adres: 43—100 Tychy, ul. Moniuszki 18/49.

Wojciech Gościniak, lat 14. Posiada mikrokomputer C--64. Zainteresowania: informatyka, fantastyka, sport i gry. Adres: 81—153 Gdynia, ul. W. Gruny 78/C/17.

Grzegorz Nalepa, lat 13. Posiada mikrokomputer Atari 800 XL magnetofon XC-12, joystick. Uzbierał około 250 programów (gry oraz użytkowe). Adres: 44—100 Gliwice, ul. Pszczyńska 112c/29.

Tomasz Witkowski, uczeń V klasy Szkoły Podstawowej. Posiada mikrokomputer Commodore 64 z zielonym monitorem drukarkę oraz stację dysków. Zainteresowania: biologia, geografia, starożytna historia, sport, muzyka, technika video, informatyka. Adres: 43—180 Orzesze, ul. Świerczewskiego 19.

Eryk Szuba, lat 15 — uczeń I klasy L.O. Posiada komputer Schneider CPC 6128 z monitorem kolorowym. Interesuje się muzyką i sportem. Proponuje wymianę oprogramowania. Adres: 04—178 Warszawa, ul. Ostrobramska 78/43.

Przemysław Ziółkowski, uczeń 13 lat. Posiada Commodore+4, magnetofon, joystick, monitor i około 700 programów (gry, edukacyjne oraz użytkowe). Proponuje wymianę programów oraz doświadczeń. Adres: 44—100 Gliwice, ul. Pszyńska 118/C/15. Adres dla korespondencji 44—101 Gliwice skr. poczt. 59 „A".

Arkadiusz Chruściel, lat 15. Posiada komputer Acorn Compact ze stacją dysków 3,5. Proponuje wymianę programów, literatury i doświadczeń. Adres: 41—200 Sosnowiec, ul. 22 lipca 94/119.

Piotr Kiszek, uczeń lat 18. Prosi o kontakt z posiadaczami mikrokomputera EINSTEIN 256. Adres: 05-120 Legionowo, ul. Olszankowa 42.

Grzegorz Mirek, lat 22. Posiada ATARI 130 XE, stację dysków 1050 oraz magnetofon 1010. Oprogramowanie gry i programy użytkowe. Proponuje wymianę oprogramowania i doświadczeń. Adres: 32-300 Olkusz, ul. Zacisze 30a.

Ludwik Nezmar, uczeń lat 16. Posiada mikrokomputer SPECTRUM 48 k z magnetofonem. Nawiąże kontakt z innymi posiadaczami tego minikomputera w celu wymiany doświadczeń i oprogramowania. Adres: 12000 Praha 2, Rumanska 9, CSSR.

Rafał Neska, uczeń lat 14. Posiada mikrokomputer C-64 z magnetofonem. Pragnie nawiązać kontakt w celu wymiany gier i programów. Adres: 01-923 Warszawa, ul. Bogusławskiego 22/69.

Andrzej Kurczewski, lat 17. Posiada Commodore 128 z magnetofonem. Oprogramowanie: gry i programy użytkowe na C-64. Proponuje wymianę. Adres: 05-300 Mińsk Mazowiecki, ul. Okrzei 18 m 4.

Wojciech Zakrzewski, lat 13. Posiada ATARI 800 XL. Zainteresowania: muzyka, gry komputerowe. Nawiąże kontakt z użytkownikami tego mikrokomputera w celu wymiany doświadczeń. Adres: 85-811 Bydgoszcz, ul. Kąkolowa 5 m 34.

Szymon Szulgacz-Noga, posiada CPC 464 z niewielką ilością oprogramowania. Interesuje się sportem oraz muzyką. Proponuje wymianę doświadczeń. Adres: 71-636 Szczecin, ul. Miedziana 13a/5.

Bartosz Hojka, uczeń 14 lat. Posiada AMSTRADA CPC 464.

KODY ASCII

32		90	Z
33	!	91	ź
34	"	92	ł
35	#	93	ń
36	$	94	ś
37	%	95	_
38	&	96	ą
39	'	97	a
40	(	98	b
41	)	99	c
42	*	100	d
43	+	101	e
44	,	102	f
45	-	103	g
46	.	104	h
47	/	105	i
48	0	106	j
49	1	107	k
50	2	108	l
51	3	109	m
52	4	110	n
53	5	111	o
54	6	112	p
55	7	113	q
56	8	114	r
57	9	115	s
58	:	116	t
59	;	117	u
60	<	118	v
61	=	119	w
62	>	120	x
63	?	121	y
64	@	122	z
65	A	123	ó
66	B	124	ę
67	C	125	ż
68	D	126	ć
69	E	127	
70	F		
71	G		
72	H		
73	I		
74	J		
75	K		
76	L		
77	M		
78	N		
79	O		
80	P		
81	Q		
82	R		
83	S		
84	T		
85	U		
86	V		
87	W		
88	X		
89	Y		

WSZYSTKO DLA WSZYSTKICH

ABM

PRZEDSIĘBIORSTWO POSTĘPU TECHNICZNEGO
SPÓŁKA Z O.O.
41-303 DĄBROWA GÓRNICZA, ULICA
CZERWONYCH SZTANDARÓW 94
TEL 64-71-48

OFERUJE:

— profesjonalny sprzęt video i TVC,
— systemy mikrokomputerowe PC/XT/AT w dowolnej konfiguracji,
— sieci lokalne i mini sieci w oparciu o PX XTV /AT,
— usługi w zakresie oprogramowania i szkolenia,
— wdrażanie systemów zarządzania,
— bogaty wybór programów użytkowych i narzędziowych,

Zapewnia:

— servis gwarancyjny i pogwarancyjny,

Wykonuje:

— usługi projektowe w pełnym zakresie budownictwa ogólnego wraz z przynależnymi dokumentacjami branżowymi,
— opinie, ekspertyzy i inwentaryzacje dla obiektów już istniejących,
— programy użytkowe i narzędziowe na zamówienie,

Skupuje:

— komputery
— sprzęt video i TVC,
— sprzęt elektroniczny wysokiej klasy, w tym: drukarki, plotery, digitizery, układy scalone, dyskietki, kable połączeniowe itp.

Naprawia:

— TVC prod. polskiej i zachodniej,
— magnetowidy,
— komputery ośmiobitowe,

Przestraja:

— TVC prod. polskiej i zachodniej,
— magnetowidy,
— radioodbiorniki zachodnie.

Informację techn.-handlową można uzyskać w siedzibie firmy "ABM"!

ATARI
ZX SPECTRUM

INSTRUKCJE
OPISY
LITERATURA

Szkoły i Kluby — Zniżka
Katalogi — Gratis
Co piąty program — Gratis
Wysyłka na cały kraj
Wypożyczalnia programów
D.H. "Sezam" II p. g. 16.00-19.00
00-849 Warszawa UPT 66, skr. p. 14.

ATARI • AMSTRAD
SPECTRUM

— instrukcje polskie
— programy użytkowe i gry
wysyła "MEGABAJT"
03-945 Warszawa, Paryska 17/29 tel. 17-76-16
— rachunki dla instytucji
— informacje po nadesłaniu koperty zwrotnej

UDOSKONALENIA
PROGRAMOWE I SPRZĘTOWE
DLA WSZYSTKICH MODELI **ATARI**
ORAZ KOMPUTERÓW PRACUJĄCYCH POD SYSTEMEM **MS-DOS**
WYSYŁA POCZTĄ
agencja mikro-komputerowa ***amico***
41-200 Sosnowiec
K-91

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

43-100 Tychy, aleja ZMP 77
tel. 27-69-75

— minikomputery 8-bitowe (Atari, Commodore, Schneider-Amstrad)
— minikomputery 16-bitowe kompatybilne z IBM PC
— drukarki 10" i 15" firm STAR, EPSON, AMSTRAD
— magnetowidy
— kamery video
— anteny satelitarne
— aparaturę badawczo-naukową

Zapewniamy o atrakcyjnych cenach.

ZX SPECTRUM
Naprawy komputerów
Interfejsy do joysticków
ATARI
Interfejs magnetofonu
Możliwość wysyłki pocztą
Jerzy Dymecki, ul. Meissnera 14 m 1, 03-982 WARSZAWA
tel. 15-93-38 po południu.

STUDIO KIJOWIANKA
AMSTRAD • ATARI XL, XE, ST • COMMODORE 64, 128
Poleca literaturę i programy na kasetach i dyskach. Warszawa, ul. Targowa 26. Rachunki oraz wysyłka pocztą. Informacje za załączeniem koperty i znaczka.

COMMODORE 16/+4/116
Wszystkie gry — tania sprzedaż — pisz: Krępna, Krótka 2
47-330 Zdzieszowice

Odstąpię programy na Commodore 64
Tomasz Jaracz
ul. Chełmińska 16/8
78-400 Szczecinek G-5

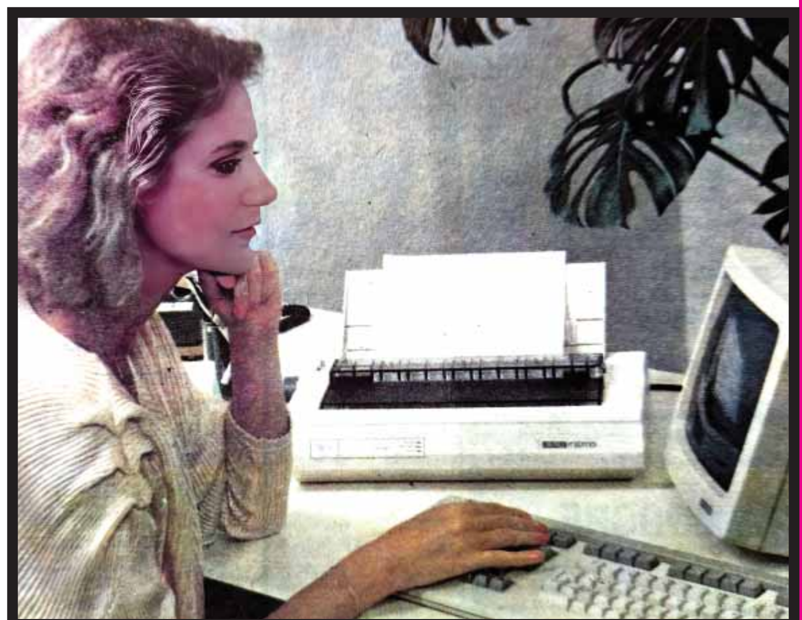
WSZYSTKO DLA WSZYSTKICH

NAJWIĘKSZY DYSTRYBUTOR PERYFERII KOMPUTEROWYCH W RFN

SYNELEC

Datensysteme
GmbH
Postfach 151727
8000 Munchen 15
tlx 5212289
tel 089/519278

CITIZEN



oferuje:

A) DRUKARKI MOZAIKOWE CITIZEN

	Cena/szt. DM	Transport DM
LSP-120D (9 igieł, 10 cali, 120 Zn/s)	350	40
MSP-15E (9 igieł, 15 cali, 160 Zn/s)	675	40
MSP-40 (9 igieł, 10 cali, 200 Zn/s)	715	40
MSP-45 (9 igieł, 15 cali, 200 Zn/s)	895	40
MSP-50 (9 igieł, 10 cali, 300/250 Zn/s)	1190	40
MSP-55 (9 igieł, 15 cali, 300/250 Zn/s)	1350	40
HOP-40 (24 igły, 10 cali, 200 Zn/s)	990	40
HQP-45 (24 igły, 15 cali, 200 Zn/s)	1450	40
Drukarka Laserowa (6 str/min.)	3990	80
Kasety do drukarek 10" (120 N)	134.5	20
10" (przy 10 szt.)	130	20
15" (przy 10 szt.)	180	20
Kable podłączeniowe do komputerów	20	

B) PLOTTERY SECONIC

SPL 410 (DIN A3)	1650	40
SPL 430 (DIN A3)	1650	40
SPL 450 (DIN A3)	2050	40
SPL 600 (DIN A2)	5900	80
SPL 800 (DIN A1)	6900	120
SPL 1000 (DIN A0)	12500	160

C) MONITOR GRAFICZNY CONRAC 19" — 5900 — 120

D) TERMINALE

ESPRiT OPUS 220	849	40
ESPRiT OPUS 2	649	40
ESPRiT OPUS 4	849	40

E) STREAMERY ARCHIVE

Archive FT 60 + cartridge	1310	40
ArchiveST600 + 1 cartridge	1310	40

F) DYSKI SEAGATE

Seagate 20 MB ST 225	399	40
Seagate 40 MB ST 251	639	40
Seagate 20 MB ST 225 z kontr, i kablem	497	40
Seagate 40 MB ST 251 z kontr, i kablem	749	40

G) DYSKIETKI MAXELL

51/4 cala		
MD1-D przy 1000 szt.	1,23	150
MD2-D przy 1000 szt.	1,58	150
MD1-DD przy 1000 szt.	1,94	150
MD2-DD przy 1000 szt.	1,99	150
MD2-HD przy 1000 szt.	3,09	150
3,5 cala		
MF1-DD przy 1000 szt.	2,62	150
MF2-DD przy 1000 szt.	2,68	150
MF2-HD przy 1000 szt.	6,54	150
3 cale		
CF2 przy 1000 szt.	4,45	150

WARUNKI HANDLOWE:

- Oferowany sprzęt zamówić można korespondencyjnie dokonując przelewu z konta A telegraficznie na nasze konto bankowe.
- Do sumy każdego przelewu prosimy dodać DM 10,- na pokrycie kosztów przelewu bankowego.
- Po dokonaniu przelewu prosimy o wysłanie do nas kopii dowodu wpłaty wraz z dokładną specyfikacją.
- Ceny należy rozumieć jako ceny z naszego składu w Munchen (FO B Munchen).
- Przy większych zakupach udzielamy rabatów.

NINIEJSZYM ZAMAWIAM:

1. .... DM ....
2. .... DM ....
3. .... DM ....

Koszty manipulacji bankowych .... **DM 10,-**

Razem DM ....

Załączam czek lub kopię zlecenia bankowego na przelew w/w sumy na konto nr 7137320 w Beyerische Vereignsbank Munchen BLZ 70020270 zrealizowanego w dniu. ....../....../....... przez bank ...., oddział......................... w ....
Podpis wpłacającego................................ Nazwisko i imie .... Data ....

NAZWISKO I IMIĘ ODBIORCY..
ADRES ODBIORCY ....

Drogi Bajtku!

Na listy czytelników odpowiada Marcin Waligórski

Od 1.5 roku jestem użytkownikiem komputera Commodora 128D ("diesel" — jak mówią niektórzy).

Dopiero całkiem niedawno odkryłem dość nieprzyjemne zjawisko: podczas pracy z liczbami zmiennoprzecinkowymi komputer... oszukuje!

Przy odejmowaniu od liczby zawierającej część całkowitą i ułamkową jej części całkowitej winien zostać ułamek postaci .xxx (kropka dziesiętna i konkretne cyfry). Tymczasem wynik jest niedokładny. Np. 212.12 - 212 = .120000005, a nie. 12.

Jeszcze ciekawsze zjawisko można zaobserwować, gdy komputer ma wyświetlić na ekranie np. liczby w przedziale od 30 do 80 z krokiem .4 (prosta BASIC-owa instrukcja FOR), i to niezależnie, czy w trybie bezpośrednim — direct mode, czy w programie. Dodaje sobie pewien mały ułamek i dla ostatniej liczby (tu: 80) w ogóle nie wykonuje działania. Zgodnie z „jego" logiką liczba 80 nie mieści się w zadanej pętli — największy element to 79.6000007. Oczywiście dla innych liczb te wartości mogą być inne np. z niedomiarem: 65.9999998 itp.

Nie ma powtarzalności np. podczas sumowania kolejnych liczb z krokiem ułamkowym, gdy następnie od takiej sumy odejmujemy te same liczby x z krokiem ujemnym. Winno się otrzymać 0, ale tak nie jest.

Miałem możliwość sprawdzenia działania innych mikrokomputerów Commodore 64/+4/116, Amstrad 464, 6128. Wyniki były identyczne! Jedynie poczciwe Spectrum dawało wyniki zgodne z oczekiwaniami. Proszę o odpowiedź na łamach „Bajtka", gdyż jak sądzę, problem wart jest omówienia..

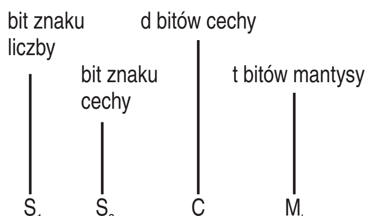
Korzystając z okazji chciałem dorzucić swoje „trzy grosze" do innych działów Waszego czasopisma: „CO JEST GRANE" — w numerze 2/88 opisano grę „HACKER" w wersji dla Atari. W przypadku Commodore wszystko się zgadza z wyjątkiem drugiego kodu rozpoznawczego — numer robota SRU wynosi AXD-0314479. Po podaniu czwartego hasła obce satelity więcej nie przeszkadzają robotowi, można bezkarnie poruszać się w dozwolonym przez program kierunku, aż do... wyczerpania limitu czasu.

Aleksander Łazowski
(adres do wiadomości redakcji)

Gratuluję spostrzegawczości i dociekliwości i spieszę wyjaśnić, choć może w pierwszej chwili zabrzmi to dziwnie, że podane błędy w działaniach na liczbach zmiennoprzecinkowych są rzeczą... normalną. Każdy, kto miał kiedyś w ręku kalkulator, zauważył, zapewne że jeżeli obliczymy np. pierwiastek z 2 i otrzymaną liczbę podniesiemy do kwadratu, to otrzymamy w wyniku nie 2 (jak być powinno) lecz np. 1.9999998. Jakkolwiek pierwiastkowanie jest niewątpliwie bardziej skomplikowanym

dla maszyny działaniem niż odejmowanie, to jednak mechanizm powstawania błędu jest ten sam.

Z czego to wynika? Z samej istoty reprezentacji liczby rzeczywistej przez komputer. Proszę bowiem zauważyć, że z samego faktu, iż do tej reprezentacji wykorzystujemy skończoną liczbę bitów, wynika, że (w ogólnym przypadku) komputer może operować jedynie na pewnych przybliżeniach tych liczb. Inaczej mówiąc, już sam zapis liczby rzeczywistej jest obarczony pewnym błędem. Aby wiedzieć, jaki to jest błąd, trzeba wiedzieć, w jaki sposób liczba jest reprezentowana. Mianowicie?



Każdą liczbę rzeczywistą, która ma zostać zapamiętana przez komputer, przekształca się do postaci

$X = S_1 * 2^c * M$, MG [Eo.5, 1)

...co pozwala nam umieścić ją w pamięci według podanego powyżej schematu. Warto zauważyć, że taka reprezentacja liczby jest dokładna wtedy i tylko wtedy, gdy mantysa M jest nieskończonej długości! Ponieważ jednak dysponujemy tylko t bitami mantysy, to x reprezentacja liczby rzeczywistej (nazwijmy ją rd (x) jest z reguły obarczona błędem.

$rd(x) = s_1 * 2^c * M$,

Łatwo zauważyć, że

$rd(x) = x * (1 + \varepsilon)$, gdzie $(\varepsilon) \leq 2^{-t}$

Inaczej mówiąc, błąd względny tej reprezentacji jest nie większy od 2^{-t}, gdzie t jest ilością bitów mantysy.

$$\frac{rd(x)-x}{x} \leq 2^{-t}$$

Skoro same liczby rzeczywiste nie są reprezentowane dokładnie to nie jesteśmy również w stanie wykonać dowolnego działania arytmetycznego z dokładnością względną lepszą od 2^{-t}. To jest właśnie przyczyną Pana wątpliwości. Jak widać z podanego przez Pana przykładu odejmowanie 2^{-t} jest w pańskim komputerze rzędu 10^{-12}.

W świetle tych faktów wcale nie dziwi, dlaczego opisana przez Pana dalej pętla nie wykonuje się dla wartości 80. Po prostu następna wyliczoną przez maszynę po 79.6000007 wartością byłoby np. 80.00000075, co jest oczywiście większe od dozorującej pętlę liczby 80.0. Przy okazji warto jednak zauważyć, że szykują się większe kłopoty: przy wielokrotnym wykonywaniu działań błędy się kumulują! Oczywiście nie musi tak być zawsze. Istnieją np. specjalne algorytmy, pozwalające na zsumowanie ciągu liczb z jednoczesnym uniknięciem kumulowania się błędów.

Nie jest wcale dziwne, że podobne zachowanie się stwierdził Pan w przypadku innych komputerów. Spectrum też nie jest wolne od tej cechy, jakkolwiek nie daje się to być może zauważyć, w tak banalnych jak podana przez Pana przypadkach. Nota bene dobrze świadczy to o twórcach arytmetyki systemowej tego komputera.

Nauki praktyczne płynące z naszych doświadczeń są następujące: po pierwsze, jako licznika pętli należy zawsze używać zmiennej typu całkowitego (obliczenia na liczbach całkowitych są dokładne!). Po drugie, najczęściej nie ma sensu porównywanie ze sobą dwóch liczb rzeczywistych w taki sposób, jak do tego przywykliśmy. Jeżeli chcemy np. sprawdzić, czy zmienna X ma wartość 0.0, to należy raczej badać czy różnica pomiędzy nią a 0.0 nie jest większa od dopuszczalnego dla niej błędu.

Powyższe rozważania są zasygnalizowaniem, a nie solidnym przedyskutowaniem reperkusji, które pociąga za sobą obarczona błędem reprezentacja liczb rzeczywistych przez maszyny cyfrowe. Całością tych zagadnień zajmuje się obszerny dział informatyki zwany analizą numeryczną. Jako literaturę polecam więc dwutomową pracę K J. i M. Jankowskich i M. Dryi „Przegląd metod i algorytmów numerycznych", Wyd. I WNT 1981. Uprzedzam jednak, że zawiera ona sporą dozę teorii i jako taka może być trudna w odbiorze.

Chcąc wprowadzić mojego 10—cio letniego syna w tematykę „Bajka" kupiłem mu komputer SHARP MZ—731 w zbudowanym magnetofonem. Szybko jednak zorientowaliśmy się, jak skąpe jest oprogramowanie do tego typu urządzenia. Spróbowałem jeszcze raz, kupując dostępny i szeroko rozpowszechniony mikrokomputer ATARI 65XE z magnetofonem. Radość trwa do chwli obecnej.

A teraz mój problem i pytania: czy prowadząc działalność handlową w sklepie z artykułami motoryzacyjnymi mogę z korzyścią dla siebie wykorzystać wymieniony wyżej mikrokomputer SHARP MZ—731 — do kogo i gdzie zwrócić się o pomoc?

Jerzy Krzemiński
ul. Artyleryjska 7/11B m 5
59—220 Legnica.

Moim zdaniem nie. Magnetofon kasetowy nie zapewnia bowiem dostatecznego bezpieczeństwa danych, a co gorsza — uniemożliwia swobodny do nich dostęp. Z tego powodu prowadzenie przy pomocy takiego komputera np. ewidencji towarów, rachunkowości czy np. mija się z celem. Informatyzacja „papierkowej roboty" ma bowiem sens wyłącznie wtedy, gdy zapis komputerowy zastępuje, a nie dubluje papiery. Papierów zaś nie można się pozbyć dopóty, dopóki nie ma się pewności, że dane nie zostaną bezpowrotnie utracone przez przypadek.

Być może wchodziłoby w grę użycie do wspomagania innych czynności, w których bezpieczeństwo zapisanych danych nie odgrywa roli. Nie jestem wprawdzie specjalistą od handlu, ale sądzę, że można pomyśleć np. o roli elektroniczej kasy (rozliczenie dzienne, druk paragonów, wykaz wpływów) czy też np. w celu druku potrzebnych formularzy, rachunków itp. Do tego wszakże potrzebna jest drukarka — a pozwolę sobie mieć wątpliwości czy osiągnięte korzyści usprawiedliwiłyby ten wydatek.

Znane redakcji firmy zajmujące się wytwarzaniem oprogramowania do celów profesjonalnych nie zajmują się komputerami nietypowymi, takimi jak SHARP (z reguły zresztą nie zajmują się komputerami domowymi — i jest to raczej zrozumiałe)

W 12 numerze „Bajtka" z ubiegłego roku został zamieszczony test drukarki TRITON. Noszę się z zamiarem zakupu jej do druku etykiet do produktów spożywczych. Przed zakupem chciałbym jednak dowiedzieć się czegoś jeszcze o tym urządzeniu:

1. Czy ta drukarka może dość długo pracować (raz na 2 miesiące przez tydzień po 10 godzin dziennie?).

2. Autorzy artykułu piszą o nietrwałości kasety z taśmą barwiącą. Czy są jakieś solidniejsze kasety do tej drukarki? Gdzie można zakupić te kasety?

3. Czy drukarka nie wymaga specjalnego papieru? Jaki można używać?

Czy można kupić to urządzenie za złotówki? Jeśli tak, to gdzie i w jakiej cenie? Jeżeli nie, to gdzie mógłbym zakupić drukarkę za dewizy i w jakiej cenie?

Jeżeli uważa Pan, że urządzenie nie nadaje się do przeznaczonego mu przez mnie celu (drukowanie etykiet), prosiłbym o polecenie innej drukarki, przydatnej do tego typu pracy.

Andrzej Bobula
ul. Padarewskiego 20/85
37—700 Przemyśl

Oto odpowiedzi dotyczące drukarki TRITON:

1. Drukarka działa bezawaryjnie pomimo intensywnej eksploatacji w redakcji „Bajtka". Ogólne wrażenie co do solidności wykonania było dobre. Trudno jednak ocenić jej trafność przy tak długotrwałej eksploatacji.

2. Niestety, wrażenie to nie dotyczyło zawartej w kasecie taśmy barwiącej. Uległa ona zużyciu w stopniu uniemożliwiającym regenerację. Sama taśma jest krótka, co dodatkowo wpływa ujemnie na jej żywotność. Ze względu na rzadko spotykany w Polsce typ kasety, należałoby ją sprowadzać z zagranicy. Redakcja nie posiada informacji o innych podobnych kasetach niż oferowane przez Radofin Elektronics.

3. Drukarka może używać każdego cienkiego papieru, którego szerokość nie przekracza 10 cali. Najlepszy jest oczywiście papier perforowany, ale dostać go w Polsce jest niezwykle trudno, w każdym wypadku zaś jest drogi. Można używać pojedynczych arkuszy A4, ale wówczas musiałby Pan zatrudnić kogoś całymi dniami do przekładania kartek papieru.... Można także użyć np. rolki papieru używanej do dalekopisu, ale wtedy też nie można pozostawić drukarki bez nadzoru, ponieważ papier ten często nie poddaje się przesuwaniu przez wałek drukarki i wskutek tego przekrzywia się.

4. Drukarka nie jest popularna w Polsce i dlatego nie potrafię wskazać miejsca, gdzie można by ją kupić. Jeżeli chodzi o zagranicę, to źródłem informacji może być firma Electronics Export, która nam drukarkę przekazała do testowania.

Moją kontrpropozycję dla wytypowanej przez Pana drukarki jest STAR SG 10. Drukarki tej serii mają ustaloną renomę co do jakości i trwałości, co w przypadku profesjonalnego zastosowania ma kapitalne znaczenie. Drukarka używa zwykłej taśmy od maszyny do pisania — co prawda nie polecałbym taśmy polskiej, ale dobre taśmę można bez trudu kupić za bony w Pewexie, znacznie poniżej kosztów jakichkolwiek kaset. Wymagania co do rodzaju papieru są równie niewygórowane jak dla TRITON--a, zaś zakładanie pojedynczych arkuszy — znacznie wygodniejsze i szybsze. Drukarki STAR są w Polsce najbardziej rozpowszechnione i najłatwiej w ich przypadku o serwis z prawdziwego zdarzenia. W chwili, gdy piszę te słowa, kilka firm oferuje SG 10 w cenie 200 funtów, ale drukarkę tę można też kupić (relatywnie drożej) za złotówki. Proszę studiować ogłoszenia w pismach poświęconych informatyce.

Zachęcam także do zrobienia dokładnej kalkulacji poniesionych na drukarkę i jej eksploatację nakładów oraz odniesionych, korzyści. Informatyzacja jest bowiem takim interesem, jak każdy inny, i dlatego jej forsowanie ma sens tylko wtedy, gdy służy obniżaniu kosztów własnych. Ze swej strony życzę powodzenia.i.

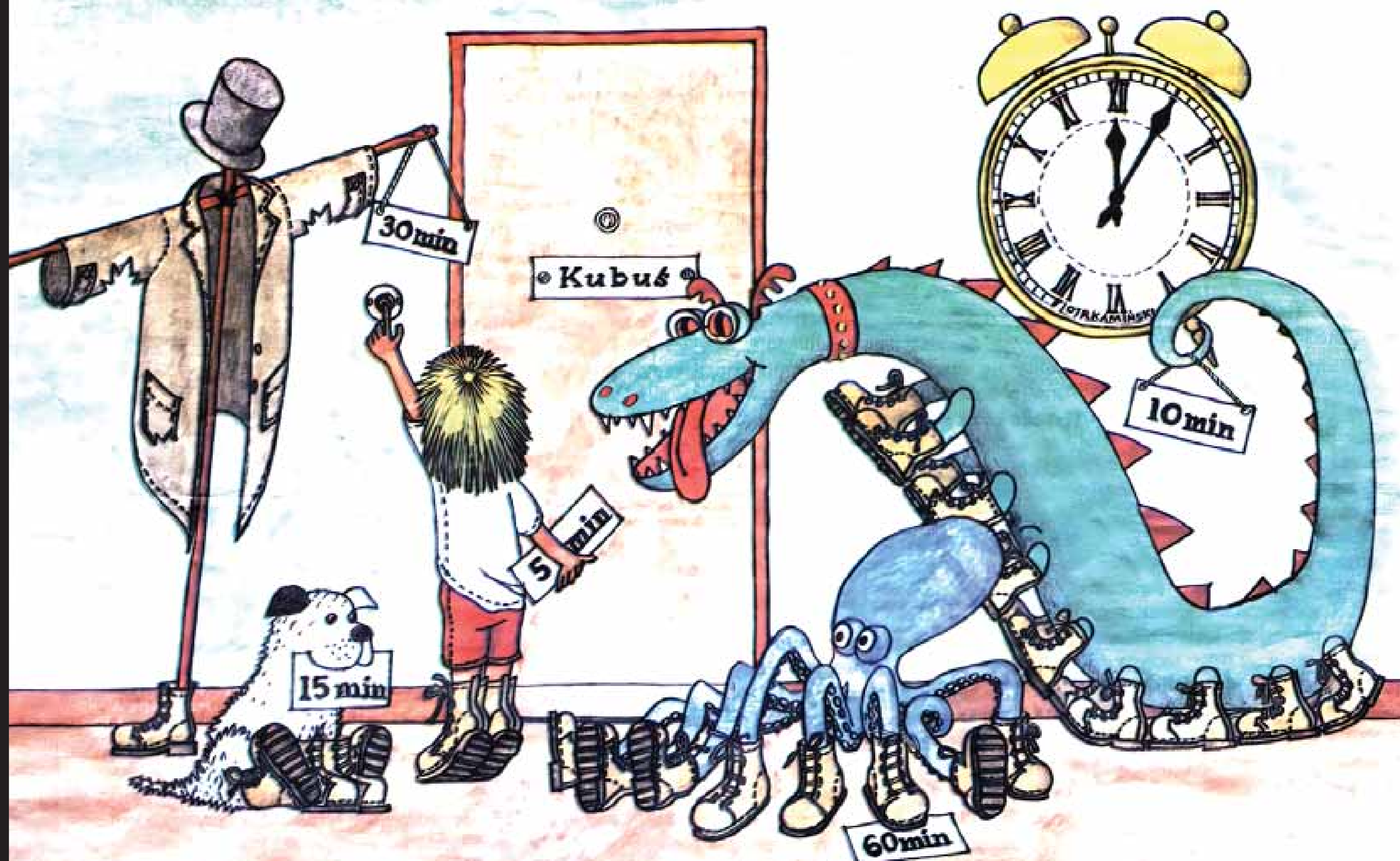
Czy mikrokomputery serii MERITUM mogą współpracować z magnetofonem, jeżeli tak, to z jakim? Czy mikrokomputery ELWRO 800 JUNIOR są dostępne w wolnej sprzedaży? Jeżeli miałby Pan kupić komputer MERITUM 1 lub 2 albo ELWRO 800 JUNIOR, to który by Pan wybrał?

Marek Zelek
ul. Klasztorna 28 m 55
33—300 Nowy Sącz

Komputery MERITUM są przystosowane do wymiany danych ze zwykłym magnetofonem.

Jak dotychczas trwają przetargi na temat zakupu przez Oświatę serii ELWRO 800 JUNIOR, wyprodukowanej już przez ELWRO. W wolnej sprzedaży mają być od czerwca.

MERITUM jest w chwili obecnej komputerem archaicznym i nie mającym żadnych perspektyw rozwoju. Wybrałbym „Juniora", ale z zastrzeżeniami. Miałem okazję pracować przez pewien czas na egzemplarzu ELWRO 800 JUNIOR pochodzącym z serii próbnej. Byłem mile zaskoczony koncepcją i możliwościami komputera, przewyższającymi moim zdaniem Amstrada 664. Natomiast jakość wykonania i trwałość samego komputera (zwłaszcza klawiatury) budziła we mnie najdalej idący krytycyzm.



KONKURS!

Kubuś Literka stał się ostatnio osobą bardzo zajętą. Drzwi jego domu nie zamykają się nawet na chwilę. Barnaba przychodzi opowiedzieć jak idzie mu w szkole, kuśtyka Strach na Wróble poskarżyć się na zuchwałe ptaki, przybiega pies Burek z najświeższymi nowinami z podwórka, przypełza ośmiornica Kornelia ponarzekać na reumatyzm i wreszcie wpada czasem sympatyczny Szesnaścior, posiadacz 16-tu nóg i wiecznie uśmiechniętego pyska.

Wystarczy wejść do przedpokoju Kubusia i rzucić okiem na szafkę z butami, by stwierdzić, jakich ma gości. Jeśli np. bucików będzie 6, to znaczy, że gośćmi są Burek i Barnaba, jeśli 11 to: ośmiornica Kornelia, Barnaba i Strach na Wróble (ten ostatni ma tylko jedną nogę, a więc nosi tylko jeden bucik).

Kubuś wie dokładnie, ile czasu zajmie mu każdy z gości. Najgorzej dogadać się z Kornelią. To straszna nudziara. Gada okrągłą godzinę. Strach na Wróble okropnie się jąka. Zanim skończy, mija pół godziny. Burek ma zawsze sporo do opowiedzenia, szczeka więc przez kwadrans. Najszybciej załatwiają sprawy Barnaba — 5 minut i Szesnaścior — 10 minut.

Jest godzina 12.00. Przychodzą pierwsi goście (szafka nr 1). Na podstawie liczby bucików możemy ustalić ich listę. Obliczamy ile czasu zajmie Kubusiowi rozmowa ze wszystkimi i zapisujemy godzinę wyjścia gości. Jest to równocześnie godzina przyjścia nowych gości szafka nr 2. Znowu obliczamy czas rozmowy, zapisujemy godzinę, itd.

Kolejne godziny i minuty są zaszyfrowanym hasłem. Kluczem do niego jest ten program. Kto nie ma komputera, będzie musiał wykonać go „na piechotę”. Zestaw kodów ASCII z umieszczonymi w nich polskimi znakami znajduje się w numerze.

Główną nagroda jest komputer ATARI 65 XE wraz z magnetofonem ufundowany przez PZ KAREN. Hasło wraz z kuponem konkursowym należy nadesłać na karcie pocztowej do dnia I 1 sierpnia 1988 roku.

Redakcja „Bajtek”, ul. Wspólna 61, 00-687 Warszawa — Konkurs z Szesnaściorem.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

KUPON KONKURSOWY

```
100 LET zestaw=0
110 PRINT "Godzina": INPUT godz
120 PRINT "minuta": INPUT mnt
130 LET l=mnt/5
140 IF godz/2<>INT(godz/2) THEN LET l=l+12
150 IF l=0 THEN LET l=24
160 IF zestaw=1 THEN GOTO 210
170 IF l=17 THEN LET zestaw=1: GOTO 110
180 IF l=22 THEN PRINT CHR$(121): GOTO 110
190 IF l=24 THEN PRINT CHR$(122): GOTO 110
200 PRINT CHR$(l+96): GOTO 110
210 IF l<7 THEN PRINT CHR$(90+l): GOTO 100
220 IF l<11 THEN PRINT CHR$(116+l): GOTO 100
230 IF l=11 THEN PRINT CHR$(32): GOTO 100
240 PRINT CHR$(52+l): GOTO 100
```

PUŁAPKA

Cześć Maluchy!

Niezbyt przyjemną niespodzianką dla każdego programisty jest znalezienie czyjegoś nazwiska w nagłówku napisanego przez siebie programu. Niestety, zdarza się to dość często. Jeśli chodzi o moje programy, to wracają one do mnie bardzo często w listach do „Bajtka" (jako własna twórczość czytelników). Żeby było jeszcze śmieszniej, są to często programy drukowane właśnie w „Bajtku".

Spróbujemy zastawić pułapkę na spryciarzy, „piratów" od siedmiu boleści próbujących przypisywać sobie cudze zasługi. Sposób zabezpieczenia jest bardzo prosty, niemniej jednak, ukryty w długim programie, powinien sprawić sporo kłopotu początkującemu włamywaczowi.

Chodzi o to, by program sam sprawdzał, czy nie dokonano w nim jakiejś zmiany. Czy jest to jednak możliwe? Okazuje się, że tak, i to w bardzo prosty sposób. Zmiana programu powoduje zmianę długości zajmowanego przez ten program obszaru pamięci, a tym samym zmienia się ilość pozostałej wolnej pamięci. Tą ostatnią wielkość możemy odczytać na większości komputerów za pomocą polecenia:

PRINT FRE (0)

Niestety, poczciwe Spectrum nie posiada tej przydatnej funkcji i w związku z tym, użytkownicy tego komputera będą sobie musieli radzić inaczej. Otóż w pamięci znajdują się dwie komórki określające gdzie kończy się program. Bez wnikania w szczegóły, możemy powiedzieć, że jeśli zmieni się długość programu, zmieni się także zawartość komórki pamięci o numerze 23627. Zawartość tej komórki możemy sprawdzić korzystając z instrukcji PEEK:

PRINT PEEK (23627)

Wystarczy więc uzupełnić program o sprawdzenie, czy ilość wolnej pamięci jest zgodna z założoną przez nas i — w przypadku stwierdzenia niezgodności,— zastosować jakąś złośliwość, na przykład wykonanie komendy NEW niszczącej cały program lub też — w przypadku pracy ze stacją dysków — komendy kasującej WSZYSTKIE PROGRAMY NA DYSKU! Dobry kawał!? Na Amstradzie można to osiągnąć przy pomocy komendy

|ERA," *.#

Cały problem w tym, że najpierw my sami musimy dowiedzieć się, jaka jest liczba wolnych bajtów pamięci. Konieczne jest również, by jej wartość nie zmieniła się po wpisaniu tejże wartości do treści programu. Wydaje się, że jest to sytuacja bez wyjścia. Długość programu zależy od liczby wolnych bajtów wpisanych w jedną z linii programu, liczba wolnych bajtów zależy zaś w sposób oczywisty od długości programu...

Na szczęście jest to tylko pozorna sprzeczność. A to z tego powodu, że liczby naturalne o tej samej długości, bez względu na wartość zajmują w pamięci tyle samo miejsca. Można więc spokojnie zastąpić naszą poszukiwaną wartość na przykład liczbą 40000.

Jest jeszcze jedna trudność. W trakcie wykonywania programu ilość wolnej pamięci zmienia się stale. Należy więc odczytać ją dokładnie w tym samym momencie, w którym zrobi to komputer. Trzeba w tym celu przerwać program. Możemy to zrobić komendą STOP lub END. Oczywiście nie będziemy mogli później tej komendy usunąć. Zamiast tego wpiszemy w linii poprzedzającej STOP rozkaz skoku. W pierwszej wersji (testowej) będzie to skok do instrukcji STOP, później zmienimy numer linii w instrukcji GOTO na następną po STOP.

Możemy więc przystąpić do pracy. Najpierw należy wybrać program, który chcemy zabezpieczyć i zastanowić się w którym miejscu umieścić pułapkę. W moim programie znalazła się ona na samym początku, ale to wcale nie znaczy, że tak być musi. Wręcz przeciwnie, im głębiej w programie zagnieździ się przygotowana przez nas „pluskwa", tym trudniej będzie ją wykryć i unieszkodliwić. Ważne jest jedynie, aby była pewność, że sprawdzenie odbędzie się tylko raz i że wcześniej nie będą stosowane żadne funkcje losowe ani też wczytywane jakiekolwiek dane (także z klawiatury i drążka sterowego). Mogłoby to bowiem spowodować przypadkową zmianę liczby wolnych bajtów.

Wróćmy jednak do naszego przykładu. W tym przypadku zabezpieczeniu będzie podlegał wielce odkrywczy i nowatorski **program na obliczanie równania kwadratowego**. (Prawda, że to wielce frapujące?) Jeśli już wklepiemy pracowicie cały listing (ten na niebieskim tle) i umieścimy w nagłówku imię i nazwisko autora, niezrównanego Kubusia Literki, musimy bardzo starannie sprawdzić, czy program działa prawidłowo i czy nie mamy ochoty czegoś w nim jeszcze zmienić. Każda zmiana dokonana później spowoduje bowiem skutki żałosne i łatwe do przewidzenia.

Nareszcie zabieramy się do zasadniczego zadania, czyli założenia pułapki. (Warto wcześniej nasz program nagrać na kasetę czy dysk). Dopisujemy do naszego programu fragment litingu na żółtym tle i po starannym sprawdzeniu poprawności uruchamiamy program. Posiadacze Spectrum muszą zmienić w nim jedną linię.

40 LET a = PEEK(23627)

Jeśli nigdzie nie popełniliśmy błędu, program powinien zatrzymać się w linii 60. Należy teraz spytać komputer, ile zostało mu wolnej pamięci. Zrobimy to w trybie bezpośrednim, to znaczy napiszemy po prostu:

PRINT a

Ja, na komputerze Amstrad 6128 uzyskałem w wyniku wartość: 1471

Możemy teraz wprowadzić uzyskaną wartość do linii **70: 70 IFa<>41471 THEN NEW**

oraz zmienić adres skoku w linii 50:

50 goto 70

Nasz program jest już gotowy i zabezpieczony. Należy go więc nagrać na taśmę lub dyskietkę. Po uzyskaniu trwałej kopii możemy spróbować eksperymentów. Na początek zmieńmy imię autora i wpiszmy np. Jakub. Okazuje się, że program uruchamia się normalnie i działa bez zarzutu. Wpiszmy teraz Barnaba..., tym razem, po uruchomieniu program sam się unicestwił. Dlaczego jednak nie zrobił tego za pierwszym razem? Czyżby wiedział, że Kubuś i Jakub to to samo imię. Oczywiście że nie! Po prostu te dwa imiona mają jednakową liczbę liter, a program może wykryć fałszerstwo tylko wówczas, gdy zmieni się liczba znaków. Tak więc opłaca się w tym przypadku projektować nagłówki z dużą liczbą informacji, aby podczas ich zmiany nasz pirat miał większą szansę wpadki. Czego życzy mu z całego serca.

Romek

```
10 REM ************
20 REM Kubus Literka
30 REM ************
40 LET a=FRE(0)
50 GOTO 60
60 STOP
70 IF a<>40000 THEN NEW
100 PRINT
110 PRINT
120 PRINT "Rozwiazanie rownania kwadratowego"
130 PRINT
140 PRINT
150 PRINT "          2      "
160 PRINT "        ax +bx+c=0"
170 PRINT
180 PRINT "Podaj, kolejne wspolczynniki"
190 PRINT
200 PRINT "a= ";
210 INPUT a
220 PRINT "b= ";
230 INPUT b
240 PRINT "c= ";
250 INPUT c
260 LET delta=b*b-4*a*c
270 IF delta=0 THEN GOTO 1000
280 IF delta<0 THEN GOTO 2000
290 LET x1=(-b-SQR(delta))/(2*a)
300 LET x2=(-b+SQR(delta))/(2*a)
310 PRINT
320 PRINT "Rownanie posiada dwa rozwiazania:"
330 PRINT
340 PRINT "x1=";x1
350 PRINT "x2=";x2
360 END
1000 x=-b/(2*a)
1010 PRINT
1020 PRINT "Rownanie posiada jedno rozwiazanie:"
1030 PRINT
1040 PRINT "x=";x
1050 END
2000 PRINT
2010 PRINT "Rownanie nie posiada rozwiazan."
2020 END
```

NIE TYLKO KOMPUTERY

Dokończenie ze str. 32

Ten zaś dla pasażerskich olbrzymów w rodzaju A320 szybko zakończyłby się zniszczeniem konstrukcji. Pierre Baud — doświadczony oblatywacz bez zmrużenia oka ciągnie joystick naszego Airbusa do siebie. Ten zaś posłuszny ręce człowieka drze w górę, po chwili jednak rezygnuje i sam wyrównuje lot. Tak nad bezpieczeństwem pasażerów czuwają elektroniczne bezpieczniki. Lotnik może popełnić błąd, a obowiązkiem komputerów jest noszenie dwóch pasów i trzech par szelek jednocześnie — opiniuje brytyjski inżynier konsorcjum Airbus-Dennis Little. Szanse, że spadną ci spodnie, czy też mówiąc prościej, że się to wszystko rozbije, są praktycznie zredukowane do zera.

Pamiętając, że w A320 zebrano rezultaty doświadczeń z niestabilnymi samolotami wojskowymi powiedzmy, że każdy nagły i silny poryw wiatru, wir wznoszący, czy tak zwana „studnia" wciągająca maszynę są elektronicznie rejestrowane i dane o nominałach trafiają do komputerów. i nim piloci zorientują się, że kaprys niebios należy natychmiast skontrować sterami, jest już po wszystkim. Automat zrobił to sam. W ten sposób każde gwałtowne czy brutalne obchodzenie się z mikrodrążkami sterowniczymi jest łagodzone przez komputery. Choć dziś trudno o tym mówić z absolutną pewnością, wydaje się, że A320 wyszedłby obronną ręką z sytuacji, kiedy podczas najbardziej złożonej fazy lotu-lądowania nagle zmienia się kierunek i siła wiatru. Przypomnijmy, iż trzy lata temu w porcie lotniczym w Dallas w USA właśnie w takich okolicznościach doszło do tragedii, która zabrała życie 113 osób. W Tuluzie na symulatorze A320 przećwiczono taki rozwój wypadków jak w 1985 roku w Teksasie, wprowadzając do systemu analogiczne dane. „Na sucho" okazało się, że nowy Airbus sprostałby takiej próbie.

Aby zrozumieć, jak to wszystko działa, spójrzmy na nie oczami dowódcy naszego A320. Ekran numer jeden — po lewej ukazuje położenie maszyny w przestrzeni względem horyzontu — wysokość — to, czy idzie w górę, czy opada. Następny wyświetla mapę nawigacyjną rejsu z zaznaczonym kursem, jego zmianami oraz radarowy obraz eteru przed nami. Następny blok zajmują standardowe (tak na wszelki wypadek) przyrządy — busola, sztuczny horyzont, radiokompas, wysokościomierz, chyłomierz i wariometr. Pośrodku natomiast jest już elektronika następnej epoki — dwa ekrany śledzące pracę silników. Nad nimi zaś — sterowany dotykiem palców — układ automatycznego pilota. Po prawej dwa pierwsze ekrany należące już do drugiego pilota.

I w ten sposób dotarliśmy do ostatecznej konstatacji — tylko dwie osoby w kabinie załogi. Bez nawigatora, radiooperatora, czy mechanika pokładowego. Wszystko jak na dłoni i w zasięgu ręki. Niektórzy złośliwi powiadają, że i tę dwójkę pilotów można byłoby skazać na bezrobocie. Co może nastąpi to w najbliższej przyszłości, ale jeszcze nie teraz. Na razie jeszcze decyduje ludzki mózg, który choć nie tak szybki jak komputer, czasami jest niezastąpiony, także i w świecie najnowszej elektroniki.

W roku bieżącym pierwsi podróżni mają szansę zakosztowania wrażeń z lotu pierwszym skomputeryzowanym odrzutowcem pasażerskim. Dostawy A320 w wersji 280-miejscowej o zasięgu ponad 5000 km dla dziesiątków towarzystw lotniczych świata powinny się już rozpocząć. W połowie 1987 roku na stale powiększającej się liście zamówień w Tuluzie komputerowa drukarka wybiła liczbę 440 egzemplarzy.

Wojciech Łuczak



LEĆ PO DRUCIE

Kiedy pochylając głowę wchodzimy do urządzonego ze smakiem i elegancją dwuosobowego kokpitu — wysuniętego w przód kadłuba miejsca pracy osób „powożących" samolotem, łapiemy w locie myśl, że ktoś z nas zakpił, że czegoś w tym wszystkim brakuje.

Ci, którzy mają jakie takie pojęcie o powietrznym rzemiośle spostrzegą po chwili, iż nie ma rzeczy podstawowej — sterownic - urządzeń przypominających wycięte kierownice samochodowe, przekazujących ruchy rąk pilotów do lotek i steru wysokości. Mniej zorientowani zdziwią się zapewne, iż ktoś zapomniał widocznie umieścić przed oczami kapitana — zajmującego lewy fotel i drugiego pilota — na prawej pozycji, tak charakterystycznych dziesiątków zegarów i wskaźników.

Właśnie z takimi wrażeniami wychodzilibyśmy z kabiny załogi pierwszego w historii lotnictwa komunikacyjnego odrzutowca, w jakim wiedza, kunszt i doświadczenie lotników wsparte są zdobyczami współczesnej elektroniki — maszyny francusko-brytyjsko- -zachodnioniemieckiej spółki (Airbus A320).

Zaczęło się od dwusilnikowego A300, który w 1974 roku rozpoczął swą karierę w barwach Air France. Jean-Yves Durance — starszy wiceprzewodniczący banku Credit Lyonnais przedstawia go w taki sposób.

— Należało zaprojektować samolot przewożący do 270 podróżnych na trasie od 28000 do 6800 km... jak najoszczędniej. I to chyba się udało. Do ubiegłego roku aż 61 towarzystw lotniczych zdecydowało się nabyć 375 A300. Za Atlantykiem u gigantów dzielących do tej pory między siebie rynek maszyn komunikacyjnych - Boeinga, McDonnella-Douglasa i Lockheeda zostało to, nie bez słuszności, odebrane jako wielkie wyzwanie.

Aby sprostać amerykańskim potentatom Airbus rzucił się w objęcia nowoczesności i nowych technologii. Jego odpowiedzią na epokę końca XX wieku jest właśnie A320.

Do momentu pierwszego wzlotu A320 w lutym 1986 r., wyobraźnia konstruktorów odrzutowych liniowców jakoś nie mogła wyzwolić się z pęt schematu - pedały orczyka i wielka sterownica ze spoczywającym na nich ciałem pilota. A wszystko to na sztywno — za pomocą stalowych cięgieł, popychaczy i całej gamy ruchowych, poddawanych zużyciu oraz przeciążeniom, elementów łączących mięśnie lotnika ze sterami — zestawione w skomplikowany mechaniczny układ. Oczywiście tam, gdzie wymagana była potworna wprost siła, dołączano hydrauliczne wspomagania, takie, jak choćby we współczesnych ciężarówkach. Tak było od osiemdziesięciu lat, od momentu, gdy bracia Wright udowodnili światu, że lot na aparacie cięższym od powietrza jest jednak możliwy! I nagle za sprawą współczesnej elektroniki standard zaczął się nagle chwiać.

Przełom rozpoczął się w lotnictwie wojskowym, kiedy po doświadczeniach wojny wietnamskiej poszukiwano super zwrotnych, gotowych do prowadzenia pojedynków manewrowych i unikania rakiet, maszyn myśliwskich. Okazało się, że najwłaściwszy byłby układ samolotu celowo źle wyważonego. Każdy kto choć raz w życiu zbudował własnoręcznie model latający szybowca wie doskonale, że próba zmuszenia do lotu niestabilnego aparatu, bez odpowiednio wybranego środka ciężkości kończy się zazwyczaj smutno. Ale niestabilny odrzutowiec bojowy, szybciej wchodzi w różnorakie zaskakujące przeciwnika ewolucje. Ba, tylko jak utrzymać go na prostej, bo przecież jakoś musi wystartować i wylądować, czy nabrać wysokości. Dla rozwiązania tego problemu zaprzęgnięto do pracy przekraczającej ludzkie możliwości zespół komputerów. Żaden, nawet najlepszy pilot nie jest w stanie błyskawicznie reagować sterami na wszelkie kaprysy niestabilnego samolotu. Szybkość reakcji naszego organizmu musi zostać wyprzedzona co najmniej parokrotnie.

Tak właśnie rodził się system sterowania odrzutowcami zwany dziś „Fly-by-wire", czyli leć po drucie, choć właściwiej byłoby rzec — lataj dzięki przewodom i to w dodatku elektrycznym. Najpierw otrzymały go maszyny wojskowe — amerykański F-16 i francuski Mirage 2000. Dziś ten układ w technice militarnej staje się rzeczą obowiązującą. Najnowsze projekty samolotów bojowych bez „Fly-by-wire" byłyby... sensacją. A 320 jest prekursorem wśród odrzutowców liniowych.

Na próżno szukaliśmy w przedziale załogowym A320 masywniej sterownicy i wolantu. Zastępuje je bowiem coś, co tak dobrze znają wielbiciele gier komputerowych — czarne, błyszczące świeżym plastikiem rączki joysticka. Przy lewej burcie kabiny dla kapitana, przy prawej dla drugiego pilota. Już w F-16 ten miniaturowy drążek przeniesiono z tradycyjnego środka, na prawo, tak aby prowadzący go lotnik operował spoczywającą w bardziej naturalnej pozycji prawą dłonią.

Każdy ruch joysticka w zelektronizowanym Airbusie wysyła określony impuls do pięciu zintegrowanych komputerów, a te analizując warunki lotu — szybkość, wysokość i parametry wiatru kierują odpowiednie rozkazy wychylającym się płaszczyznom sterowym. Podatny na uszkodzenia układ mechaniczny sterowania został ograniczony do minimum. Przewody elektryczne — „druty" zastąpiły cięgna i siłowniki. A w przypadku niestabilnych myśliwców, zespół komputerowy bez wiedzy człowieka zawiaduje szybkimi jak mgnienie oka ruchami sterów sprawiającymi, że coś, co powinno wić się w powietrzu na podobieństwo węża, leci o dziwo, jak po sznurku. Jest to tak zwany zespół równowagi dynamicznej.

Ale wróćmy do kokpitu naszego A320. Szef pilotów doświadczalnych spółki Airbus chętnie demonstruje w powietrzu walory inteligentnego samolotu. Ciągnąc do siebie drążek konwencjonalnej maszyny zadzieramy jej nos. Wejdzie więc we wznoszenie. Po jakimś czasie jednak utraci całkowicie prędkość w stromej świecy. Wówczas standardowo zbudowany samolot zgodnie z zasadami autorotacji powinien rozpocząć korkociąg.

Dokończenie na str. 31



Na razie jeszcze decyduje ludzki mózg...



www.t2e.pl

Seria „Reduks" ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków", jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe" są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.

WSZYSTKO DLA WSZYSTKICH